

From Counting to Systems of Equations

A compositional and higher-categorical semantics for an educational game for ages 4-10

Anonymous

15 October 2026 · Manuscript version 1.0

Abstract

We specify Number Garden, an educational game in which finite collections, equal groups, unknown quantities, simultaneous equations, and short stories share a compositional semantics. A typed language of natural-number quantities supplies denotations; a transition system separates reversible mathematical rewrites, information-revealing actions, tentative assignments, and physical changes. We prove counting correctness, substitution, preservation of solution sets, soundness of accepted answers, correctness of a controlled-story compiler, and completeness of bounded solution enumeration. A presheaf of solutions describes the acquisition of clues. A groupoid-valued refinement records permutations of counting tokens, and its objectwise 0-truncation recovers the numerical semantics. A gluing theorem states the precise compatibility and constraint-coverage conditions under which partial solutions combine. Worked tasks include two equations in two unknowns and three equations in three unknowns. A dependency-free reference implementation checks the finite semantics and counterexamples. The contribution is a mathematical design and executable specification, not evidence that this untested game improves learning.

1 Purpose, scope, and evidence

An educational game should connect an action to a mathematical reason. Moving three counters beside two counters should express addition. A mystery bag should denote a quantity that remains the same wherever its label occurs. Removing matching groups from a balance should preserve the possible answers. A story should supply constraints whose meaning survives its translation into an equation.

This paper specifies **Number Garden**, a cooperative game in which a child helps characters count, distribute, and identify collections of objects. Its intended audience spans approximately ages 4-10. These ages guide interface design; they are not prerequisites, diagnostic categories, or promises of attainment. Simultaneous equations are an optional later strand, introduced through small concrete puzzles after the relevant earlier skills. A four-year-old can use the counting activities without encountering symbolic elimination.

The design has four layers: a child-facing activity, a mathematical state, a small trusted checking kernel, and an instructional policy. The policy selects examples and assistance. The kernel decides whether a mathematical claim follows from the displayed problem. A recommendation system, language model, or scoring heuristic cannot override the kernel.

The progression is informed by the IES practice guide on early mathematics [1], numerical-board-game research [2], early-algebra intervention research [3], and the counting, equality, unknown-number, and multistep-problem strands of the Common Core standards [4-6]. These references support considering the component ideas. None evaluates Number Garden. In particular, the systems-of-equations strand is a proposed enrichment sequence, not a claim that simultaneous linear equations are required by the cited elementary standards. The contribution is the integra-

tion of familiar mathematics into explicit semantic contracts; no priority claim is made for its elementary theorems.

2 Game design and progression

Each episode has a scene, a mathematical question, available actions, feedback, and an exit. A child may hear the story, manipulate its objects, request a hint, offer an answer, explain a move, or stop. Quantity is encoded by countable objects and numerals; bag identity is encoded by shape, label, and optional color together. Audio repeats the same approved story. No mathematical distinction depends on color alone, reading speed, or response time.

Strand	Illustrative entry range	Child-facing task	Mathematical target
Collections	4-6	Touch each seed once; match seeds to pots	One-to-one correspondence, cardinality, zero
Join and separate	5-7	Join two trays; give away part of a collection	Addition; subtraction as an unknown addend
Groups and equality	6-8	Build equal rows; compare two sides of a balance	Multiplication by a known number; equality
Hidden quantities	6-9	Fill a mystery bag; complete either side of an equation	Variables, repeated occurrences, inverse reasoning
Linked clues	8-10	Find the contents of two differently labeled bags	Conjunction of equations; substitution and cancellation
Story investigations	8-10, optional extension	Combine three clues; detect missing or conflicting information	Three or more unknowns; modeling and checking

The intervals overlap deliberately. Entry is based on observed task performance and adult judgment. A child can return to concrete objects at any stage. Two unknown bags appear together before letters are introduced. The same shape and label denote the same unknown across the scene; distinct labels denote independent quantities unless a clue relates them.

A typical episode is: listen or read; predict; build or transform a model; check the complete answer against every clue; explain one relationship; encounter a new presentation of that relationship. Incorrect moves produce specific feedback and leave the accepted mathematical state intact. Stars, narrative rewards, and animations may celebrate participation, but the state `solved` is reserved for the checker described below.

3 Quantities, syntax, and denotation

Definition 1 (quantity context). A context is a finite ordered list

$$\Gamma = (x_1 : u_1, \dots, x_m : u_m),$$

where the variables are distinct and u_i is a unit such as seeds or beads. A natural-number valuation assigns $\rho(x_i) \in \mathbb{N}$ to each variable. The empty collection has cardinality zero, so $\mathbb{N} = \{0, 1, 2, \dots\}$. For a finite puzzle, declare domains $D_i = \{0, \dots, B_i\}$ and set $V_\Gamma = \prod_i D_i$. Bounds apply to unknown values; arithmetic uses ordinary, unbounded natural numbers, without wraparound or saturation.

Definition 2 (typed linear terms). For each unit u ,

$$t : u ::= n_u \mid x : u \mid (t : u) + (t : u) \mid k(t : u), \quad n, k \in \mathbb{N}.$$

An equation $t = s$ is well typed only when both sides have the same unit. Scalar multiplication kt means k equal groups of the quantity denoted by t . It does not multiply two unknown quantities. Multiplication and exact division of known whole-number quantities are included: for example, $3 \cdot 4 = 12$ and the question $3x = 12$. Fractions, negative numbers, and nonlinear equations are outside the core language.

Mixed-unit additions are rejected. A story about rabbits and birds must explicitly introduce a common counting unit, animals, before summing their counts. Counting containers and counting their contents are also different: the sentence “three bags, each with x seeds” contributes $3x$ seeds, not $3 + x$.

Define evaluation recursively:

$$\llbracket n_u \rrbracket_\rho = n, \quad \llbracket x \rrbracket_\rho = \rho(x), \quad \llbracket t + s \rrbracket_\rho = \llbracket t \rrbracket_\rho + \llbracket s \rrbracket_\rho, \quad \llbracket kt \rrbracket_\rho = k \llbracket t \rrbracket_\rho.$$

Definition 3 (problem and solution). A problem is $P = (\Gamma, D, E, q)$, where E is a finite conjunction of well-typed equations and q is a declared question. Its solution set is

$$\text{Sol}(P) = \{\rho \in V_\Gamma : \llbracket t \rrbracket_\rho = \llbracket s \rrbracket_\rho \text{ for every } (t = s) \in E\}.$$

Three question modes are distinguished. **Find** asks for any member of this set. **Determine** asks for the value that a target term takes in every solution and requires the solution set to be nonempty. **Classify** asks whether there are zero, one, or several solutions, optionally requesting witnesses. A standard mystery-bag episode requires a unique complete assignment. This is a generation contract, not a consequence of having as many equations as variables.

Subtraction is represented relationally: “ a objects before, b removed, c left” means $c + b = a$. Thus $b > a$ has no natural-number solution. There is no silent replacement of a negative result by zero. Exact division is $kx = n$ with $k > 0$; a nonzero remainder is a different, explicitly declared problem, expressible as $kx + r = n$ with a finite domain $0 \leq r < k$.

Lemma 1 (substitution). Let σ replace each variable x by a same-unit term $\sigma(x)$ over a context Δ . For every natural-number valuation η on Δ , define $\sigma^*\eta(x) = \llbracket \sigma(x) \rrbracket_\eta$. Then

$$\llbracket t[\sigma] \rrbracket_\eta = \llbracket t \rrbracket_{\sigma^*\eta}.$$

Proof. The constant and variable cases follow from the definitions. The addition case follows by applying the induction hypothesis to both summands. The scalar case follows by multiplying the

induction hypothesis by k . These are all constructors. For finite-domain problems, the induced valuation is admissible only if every substituted value belongs to its declared domain. $\square$

Terms modulo the commutative-monoid laws, with a distinguished unit quantity 1_u , form a many-sorted algebraic syntax. Tuples of terms compose by substitution; evaluation is a product-preserving interpretation in sets. This is an application of functorial semantics [7]. Cancellation is a property of the chosen natural-number model; it is not silently added as a law of every commutative monoid.

4 Counting as a witnessed operation

Definition 4 (counting certificate). A scene contains a finite set A of individually identifiable tokens. A certificate for the answer n is a bijection $c : [n] \rightarrow A$, where $[n] = \{1, \dots, n\}$. Operationally, the child selects tokens; a selected token cannot increment the count again. Completion requires that every token has been selected.

Theorem 2 (counting correctness and invariance). A valid certificate proves $n = |A|$. If $f : A \rightarrow A'$ is a bijection, then $f \circ c$ certifies the same number for the rearranged scene. If A and B are collections, then

$$|A \sqcup B| = |A| + |B|, \quad |[k] \times A| = k|A|.$$

Proof. A bijection to $[n]$ is the definition of finite cardinality. Composing bijections preserves bijectivity. Concatenating enumerations of the tagged disjoint union gives the addition formula; enumerating k tagged copies gives the multiplication formula. $\square$

Tagging matters. Joining two overlapping selections of the same physical objects does not automatically create their disjoint union. A valid join either proves disjointness of existing token identifiers or creates explicitly distinct copies. Repositioning a token preserves its identifier. This prevents accidental double counting when animations overlap or an object disappears behind another.

Let $\mathbf{Fin}^\approx$ be the groupoid of finite sets and bijections. Its isomorphism classes are $\mathbb{N}$, with disjoint union and product inducing addition and multiplication. This groupoid retains permutation witnesses that cardinality forgets. It is not itself asserted to be a topos, and its cardinality interpretation is not a faithful functor.

5 Operational semantics and mathematical safety

Definition 5 (game state). A state is $s = (P_0, E, \alpha, R, h)$: the immutable original problem P_0 , the currently displayed equations E , a tentative partial assignment α , a representation R such as counters or bags, and a finite history h . In the core equation mode, the context and domains remain fixed. An accepted rewrite must preserve the solution set of E . Partial assignments are proposals; they do not change the original problem or become deductions merely by being entered.

An action has one of four meanings:

1. **Rewrite:** a checked equivalence $\text{Sol}(E) = \text{Sol}(E')$.
2. **Reveal:** expose a previously authored clue e , replacing E by $E \cup \{e\}$. This can narrow the solution set; the interface must identify it as new information.
3. **Try:** change α and test consistency against the current clues. Inconsistent guesses remain unaccepted proposals and can be revised.

4. **World change:** add, consume, or move quantities between time-indexed locations. This creates a new relational problem, for example $a_{t+1} + 2 = a_t$, rather than claiming that the original quantity is unchanged.

A verified world change may conserve the total across locations while changing an individual location's count. This is different from preserving the solution set of a fixed equation presentation.

Theorem 3 (sound rewrite rules). Each of the following preserves the solution set over fixed natural-number domains:

- (a) replacing a term by an associatively or commutatively regrouped term, expanding $k(t + s)$ to $kt + ks$, and evaluating closed arithmetic;
- (b) replacing $t = s$ by $t + r = s + r$, or conversely;
- (c) replacing $kt = ks$ by $t = s$, or conversely, when $k > 0$;
- (d) substituting equals into another equation while retaining the equation that justifies the substitution;
- (e) appending a derived equation while retaining its premises.

Proof. For each admissible valuation, (a) follows from natural-number arithmetic. In (b), addition by $\llbracket r \rrbracket_\rho$ is injective. In (c), multiplication by a positive integer is injective. For (d), the retained equality makes the replaced subterms evaluate equally; induction on the containing term supplies both directions. For (e), every original solution satisfies the consequence, while retaining the premises prevents new solutions. Conjoin the unchanged equations in each case. $\square$

The side conditions are essential. Multiplication by zero can turn a false equality into $0 = 0$. Replacing two equations by their sum can lose information. Removing an unequal number of tokens from the sides can change the answer. A general instruction to “do the same thing to both sides” is valid only when the operation is reversible on the relevant values, or when the remaining constraints separately establish equivalence.

Lemma 4 (safe elimination of a variable). Suppose $E = \{x = t\} \cup F$, x does not occur in t , and x has domain D_x . Projection away from x is a bijection from $\text{Sol}(E)$ to the valuations η satisfying both $F[t/x]$ and $\llbracket t \rrbracket_\eta \in D_x$.

Proof. A solution projects to such an η by Lemma 1 and the bound on x . Conversely, extend η by the unique value $x = \llbracket t \rrbracket_\eta$. It satisfies the bound, the defining equation, and F by Lemma 1. The constructions are inverse. $\square$

The bound cannot be dropped: with $x, y \in \{0, 1\}$ and $x = y + 1$, only $y = 0$ extends. Eliminating x and leaving an unconstrained y would introduce $y = 1$. The reference runtime avoids this hazard by retaining the original context and equations during ordinary play.

Theorem 5 (no false mathematical success). Assume the initial problem is well typed, every accepted rewrite satisfies Theorem 3 or an independently checked equivalent finite-domain certificate, and the success checker evaluates the immutable original problem. Then a **Find** success returns an actual solution of P_0 . A **Determine** success returns a value shared by every solution of P_0 , with at least one such solution.

Proof. Induction on accepted rewrites preserves the solution set. A **Find** certificate is a complete in-domain assignment passing every original equation. A **Determine** certificate is a nonempty enumerated solution set whose target-value image is the singleton containing the answer. Both conditions are exactly their respective semantics. Reveals do not remove original authored clues from the final check; guesses are never treated as premises. $\square$

This theorem concerns accepted mathematics. It does not imply that a child understood a successful move. Guessing a valid answer, following a demonstration, and independently deriving an answer are recorded as different forms of instructional evidence.

6 Word problems as a small relational language

Definition 6 (controlled stories). A story has declared quantities and units, a question, and a conjunction of clauses built from:

Clause	Intended relation	Compiled equation
<code>join(a,b,c)</code>	Disjoint quantities a, b form c	$a + b = c$
<code>more(a,b,n)</code>	a is n greater than b	$a = b + n$
<code>groups(k,a,b)</code>	k equal groups of a form b	$ka = b$
<code>remove(a,b,c)</code>	From a , remove b , leaving c	$c + b = a$
<code>same(a,b)</code>	The two quantities are equal	$a = b$

Every entry is unit checked; k, n are known natural numbers. In `join`, disjointness is a scene-authoring obligation. In `remove`, quantities at different times receive different identifiers. A vocabulary renderer can say “two more than,” “two fewer than,” or a translated equivalent only when its orientation is explicitly validated against the clause. The mathematics is not inferred from an isolated keyword such as “more.”

Theorem 6 (compiler adequacy). Give each controlled clause the intended relation in the table, and interpret story conjunction as simultaneous satisfaction. For any well-typed story W and admissible valuation ρ ,

$$\rho \models W \iff \rho \in \text{Sol}(C(W)),$$

where C is the table-driven compiler.

Proof. Each atomic case is the corresponding relation in the table. The conjunction case follows because the compiler concatenates equation lists and both interpretations require every conjunct. Variable binding and units are preserved by the declarations. Structural induction proves the claim. $\square$

This is an adequacy result for an explicitly defined controlled language. It is not a theorem about understanding unrestricted English. Human-authored and generated stories require a review that the displayed wording, image, audio, referents, time labels, and intended clause agree. A language model may propose wording; it supplies no proof that children will interpret it correctly.

7 Worked episodes: one, two, and three unknowns

Episode A: counting and an unknown addend. Five seeds are displayed with stable identifiers. Selecting each exactly once certifies five. Next, three seeds are visible and a closed bag completes a total of eight: $3 + x = 8$. Move a matched group of three to each side’s explanation tray. The remaining collection certifies $x = 5$. This introduces an unknown through a collection before requiring a letter.

Episode B: two bags and two clues. A triangle bag and a square bag together contain eight beads. The triangle bag contains two more beads than the square bag. With x and y their respective counts, and $D_x = D_y = \{0, \dots, 10\}$,

$$x + y = 8, \quad x = y + 2.$$

Substitute the second clue into the first and regroup:

$$(y + 2) + y = 8, \quad 2y + 2 = 6 + 2, \quad 2y = 6, \quad y = 3.$$

The defining clue gives $x = 5$. Every displayed step retains enough premises to preserve the full solution set. Children can instead try pairs systematically, using a split tray with eight beads. Final feedback checks both $5 + 3 = 8$ and $5 = 3 + 2$; satisfying only the total is insufficient.

Episode C: equal groups and a second unknown. Two triangle packets and one square packet contain eleven seeds. One triangle packet and one square packet contain seven seeds. Each occurrence of a packet label has the same content count:

$$2x + y = 11, \quad x + y = 7.$$

Rewrite the first left side as $x + (x + y)$, use the second clue, and obtain $x + 7 = 11 = 4 + 7$. Cancellation yields $x = 4$, followed by $y = 3$. This can be enacted as comparing two arrangements and identifying the extra triangle packet. It requires no negative tokens.

Episode D: three bags. Pairwise clues give

$$x + y = 7, \quad y + z = 5, \quad x + z = 6, \quad x, y, z \in \{0, \dots, 10\}.$$

Append the sum of the first and third equations, retaining both: $2x + y + z = 13$. Substitute the middle clue to get $2x + 5 = 13 = 8 + 5$, hence $2x = 8$ and $x = 4$. The remaining clues give $y = 3$ and $z = 2$. Substitution into all three original equations certifies the answer. The example demonstrates arbitrary finite contexts beyond two unknowns without requiring all learners to enter this strand.

Episode E: learning to ask for a clue. The equations $x + y = 8$ and $2x + 2y = 16$ have nine solutions in $\{0, \dots, 10\}^2$. Two equations therefore do not determine two unknowns. The appropriate answer to “what must x be?” is that the clues do not determine it. The total $x + y$, however, is determined. Conversely, $x + y = 8$ and $x + y = 9$ are inconsistent. These episodes are explicitly labeled investigations; they are not mislabeled as unique-answer tasks.

8 Constructing and checking puzzles

Theorem 7 (finite decision procedure). Enumerating V_Γ and retaining exactly the valuations satisfying E terminates and returns exactly $\text{Sol}(P)$. It therefore decides existence, uniqueness, consistency of a partial assignment, and determinacy of a target term.

Proof. The product has $N = \prod_i (B_i + 1)$ elements. Enumeration visits each tuple once; term evaluation terminates by structural recursion. A tuple is retained if and only if it satisfies every defining equation. Counting the retained tuples and evaluating the target decide the listed properties. $\square$

If the total expression-tree size of the equations is L , the procedure uses $O(NL)$ arithmetic operations. This is an operation count, not a bit-complexity bound for arbitrarily large coefficients. Checking a proposed complete answer uses $O(L)$ operations. For three unknowns bounded by ten, there are only $11^3 = 1331$ candidate tuples. A larger deployment should use pruning or a verified constraint solver while retaining the same semantics.

A generator first chooses an in-domain witness ρ_* , then constructs clues true of it, and finally enumerates to enforce the selected question mode. For **unique**, the entire solution set must be $\{\rho_*\}$. Generation by a witness ensures existence but does not ensure uniqueness. The generator separately bounds displayed totals, group sizes, reading demands, and the length of a demonstrable solution. These instructional filters are not consequences of satisfiability.

For the particular system $x + y = s$, $x = y + d$ with $d \geq 0$, a natural-number solution exists exactly when $s \geq d$ and $s - d$ is even, subject also to the declared domains; then $y = (s - d)/2$ and $x = (s + d)/2$. Choosing y, d first provides integral answers without post hoc rounding. Full

column rank over $\mathbb{Q}$ guarantees at most one rational solution to a linear system, but not existence of an in-domain natural-number solution. Rank deficiency does not preclude uniqueness on a restricted nonnegative domain: $x + y = 0$ forces $x = y = 0$.

9 Clues, presheaves, and higher structure

Fix a finite context, its domains, and a finite set E_* of authored clues. Let $\mathcal{C} = \mathcal{P}(E_*)$ be the finite poset category, with an arrow $K \rightarrow L$ when $K \subseteq L$. Define

$$S : \mathcal{C}^{\text{op}} \longrightarrow \mathbf{Set}, \quad S(K) = \{\rho \in V_\Gamma : \rho \models K\}.$$

For $K \subseteq L$, restriction $S(L) \rightarrow S(K)$ is inclusion. Revealing a clue traverses the information order $K \subseteq L$ and reduces the candidate set. It is not generally a function that sends every earlier candidate to a later candidate. This variance distinguishes new information from an invertible rewrite.

Proposition 8 (solution presheaf). The displayed assignments define a presheaf and a monomorphism $S \hookrightarrow \Delta V_\Gamma$ into the constant presheaf. Its global sections are naturally in bijection with $S(E_*)$.

Proof. Inclusions compose and preserve identities. Each component into V_Γ is injective and commutes with restriction, proving the monomorphism. A global section is a compatible family ρ_K . Compatibility with $K \subseteq E_*$ forces every ρ_K to be the same tuple as ρ_{E_*} . Conversely, any tuple satisfying all clues supplies that family. $\square$

The ambient category $\widehat{\mathcal{C}} = \mathbf{Fun}(\mathcal{C}^{\text{op}}, \mathbf{Set})$ is the ordinary presheaf topos. The solution presheaf is an object of it, not itself a topos. A valuation is an element at a stage; it should not be confused with a geometric point of the ambient topos. No nontrivial Grothendieck topology on $\mathcal{C}$ is imposed.

For a higher refinement, let $\mathcal{S}$ denote the ∞ -category of spaces and define the presheaf ∞ -topos

$$\mathcal{X} = \mathbf{Fun}(N(\mathcal{C})^{\text{op}}, \mathcal{S}).$$

This is the standard presheaf construction in higher topos theory [8]. For each solution, retain the symmetries of its abstract counting sets:

$$\widetilde{S}(K) = \coprod_{\rho \in S(K)} \prod_{i=1}^m B\Sigma_{\rho(x_i)},$$

where Σ_n is the permutation group of n objects and $B\Sigma_n$ its one-object groupoid, regarded as a space by its nerve. The groups for zero and one object are trivial. These are abstract relabeling symmetries; the runtime retains stable physical identifiers.

Proposition 9 (recovery of numerical semantics). The inclusions of solution-indexed summands make $\widetilde{S}$ an object of $\mathcal{X}$. It is 1-truncated, and its objectwise 0-truncation satisfies $\tau_{\leq 0} \widetilde{S} \simeq S$.

Proof. Restriction keeps the same solution tuple and acts identically on its factors, so the presheaf laws hold. Every $B\Sigma_n$ is a connected 1-type. Finite products remain connected 1-types; coproducts have one connected component per solution. Taking connected components therefore yields exactly $S(K)$, naturally in K . $\square$

This is a conservative higher representation of the arithmetic answers: it adds witnesses and symmetries without adding solutions. It does not assert that the truncation preserves all limits, or that it is a left-exact localization. It also does not identify all solution strategies. Operational

histories are directed paths in the transition graph. Equal final answers can arise from different histories, and those histories carry different instructional evidence. No groupoid completion of those paths is used to justify undoing a world change.

10 Compatibility and the gluing of partial solutions

There is a second, distinct indexing space: the discrete space of variables $I = \{1, \dots, m\}$. For $U \subseteq I$, put $A(U) = \prod_{i \in U} D_i$. Restrictions forget coordinates. This assignment sheaf glues compatible partial valuations by ordinary coordinatewise union.

Theorem 10 (constraint-aware gluing). Let $U_1, \dots, U_r$ cover I . Assign each equation $e \in E$ to at least one U_j containing all variables occurring in e ; denote the equations assigned to U_j by E_j . Suppose $\rho_j \in A(U_j)$ satisfies E_j , and ρ_i, ρ_j agree on $U_i \cap U_j$. Then there is a unique $\rho \in \text{Sol}(P)$ restricting to all ρ_j . Conversely, each global solution produces such a compatible family.

Proof. Set $\rho(x_i) = \rho_j(x_i)$ for any U_j containing i . The cover gives existence and agreement on overlaps gives independence of the choice. This also proves uniqueness. Each equation is evaluated entirely in a set to which it was assigned, where it holds. Conversely, restrictions of a global solution agree and satisfy every assigned local equation. $\square$

Both hypotheses do work. Locally solvable clues $x = 1$ and $x = 2$ cannot be combined with incompatible assignments to the shared variable. Moreover, covering the variables alone does not cover their relationships. For $E = \{x = y\}$ and the cover $\{x\}, \{y\}$, the assignments $x = 0$ and $y = 1$ agree on their empty overlap but violate the omitted cross-variable constraint. Equivalently, the presheaf of restrictions of global solutions need not be a sheaf: its singleton restrictions allow this incompatible numerical combination. The theorem does not assert unconditional descent of arbitrary puzzle solutions.

In a cooperative game, each child can work with a subset of clues, but the system must check shared quantities and all cross-group constraints before declaring the joint answer solved.

11 Instructional adaptation and evaluation

Let $(K, \preceq)$ be a finite partial order of skills, where $a \preceq b$ means that a is a declared prerequisite of b . An instructional record $M \subseteq K$ is maintained as a downward-closed set. A new skill b is eligible when its strict predecessors belong to M . A local policy may request several independent successes, an explanation, a delayed revisit, or an adult decision before adding b .

Proposition 11 (prerequisite preservation). If M is downward closed and every strict predecessor of b lies in M , then $M \cup \{b\}$ is downward closed.

Proof. For an element already in M , the claim follows from downward closure. For b , every lower element is either b itself or an assumed predecessor. $\square$

This proposition verifies a curriculum data invariant. It does not prove that the record measures mastery accurately. Success, hesitation, a language difficulty, and motor access needs are not interchangeable indicators of mathematical understanding. The instructional record should preserve the provenance of evidence: assisted versus independent, familiar versus novel presentation, and immediate versus delayed performance. The proposed design uses no competitive speed ranking and makes no reward conditional on disclosing personal information.

Evaluation must therefore have two separate tracks. Formal evaluation checks definitions, proofs, compiler contracts, solution enumeration, and runtime invariants. Educational evaluation would begin with adult usability review and supervised formative sessions, then a prospectively specified comparative study. Suitable outcomes include cardinality under rearrangement, interpretation of equality, transfer to unseen missing-number and story tasks, retention, and explanations of linked

clues. Comparison activities should have comparable instructional time and adult assistance. Analyses should distinguish starting skill, language access, and assistance. This paper reports no participants, effect sizes, classroom observations, or claims of developmental validation.

12 Reproducibility, limitations, and authorship

The accompanying source bundle contains the manuscript, a typeset PDF, a self-contained mathematical HTML rendering, a journal-length submission text, and a dependency-free Python reference kernel. The kernel implements typed linear expressions, finite domains, controlled-story compilation, solution enumeration, answer checking, and checked rewrite equivalence. Its verification script passed 15,619 checks, including 1,296 small matrix systems, 66 generated two-bag puzzles, 874 complete counting orders, and negative cases for unsound transformations. The generated verification report records the exact finite grids and counts through its companion script. These checks support the executable finite model; the general theorems above are prose proofs, not Lean, Coq, or Agda developments.

The prototype is a mathematical reference implementation, not a deployed child-facing game. It does not implement speech recognition, assess unrestricted explanations, validate a browser's token rendering, or certify learning outcomes. The correctness claims assume that displayed scenes faithfully represent the authored clauses and that the trusted checker is executed as specified. A production implementation must separately establish this rendering and interaction contract. The bounds and linear syntax are explicit; no claim of completeness for unbounded Diophantine reasoning is made.

Authorship is recorded under an anonymous human byline for this version. OpenAI Codex assisted in drafting, source retrieval, mathematical analysis, typesetting, and the reference implementation. The commissioning author retains responsibility for approving the final text and attribution. Tool output is not treated as a referee report or as an authority replacing the stated arguments. This manuscript has not been accepted or peer reviewed merely by being prepared for Higher Categorical Semantics.

13 References

- [1] D. Frye, A. J. Baroody, M. Burchinal, S. M. Carver, N. C. Jordan, and J. McDowell. *Teaching Math to Young Children*. NCEE 2014-4005, Institute of Education Sciences, 2013. [Official guide and evidence ratings](#). Used for developmental progression and attention to children’s existing knowledge; the guide’s recommendations have different evidence ratings.
- [2] G. B. Ramani and R. S. Siegler. Promoting broad and stable improvements in low-income children’s numerical knowledge through playing number board games. *Child Development* 79(2), 375-394, 2008. [doi:10.1111/j.1467-8624.2007.01131.x](#). [Verified abstract and bibliographic record](#). Evidence concerns the studied numerical board game, not the present design.
- [3] M. Blanton, A. Stephens, E. Knuth, A. M. Gardiner, I. Isler, and J.-S. Kim. The development of children’s algebraic thinking: The impact of a comprehensive early algebra intervention in third grade. *Journal for Research in Mathematics Education* 46(1), 39-87, 2015. [doi:10.5951/jresmetheduc.46.1.0039](#). [Author-institution record and abstract](#). Supports the feasibility of investigating early algebra with appropriate instruction; it does not establish an age threshold for the proposed game.
- [4] National Governors Association Center for Best Practices and Council of Chief State School Officers. *Common Core State Standards for Mathematics*, 2010. [Kindergarten: Counting and Cardinality, especially K.CC.B.4](#).
- [5] Same authors. [Grade 1: Operations and Algebraic Thinking, especially 1.OA.A.1 and 1.OA.D.7-8](#).
- [6] Same authors. [Grade 4: Operations and Algebraic Thinking, especially 4.OA.A.1-3](#).
- [7] F. W. Lawvere. *Functorial Semantics of Algebraic Theories* (1963), with *Some Algebraic Problems in the Context of Functorial Semantics of Algebraic Theories* (1968). Reprinted in *Reprints in Theory and Applications of Categories* 5, 1-121, 2004. [Publisher record](#); [full text](#).
- [8] J. Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170, Princeton University Press, 2009. [Author-hosted text](#); [arXiv record](#). See Chapters 5-6 for presheaf categories, higher topoi, and truncation. The specific solution presheaves and symmetry refinement in this paper are constructed explicitly above.

Reference URLs and the journal’s public interface were checked on 15 October 2026. Bibliographic verification does not imply a systematic review of all relevant educational research.