


# To Flatten a Stone with Six Birds: Critical Pairs, Holonomy, and Confluence in Rewriting Systems

Ioannis Tsiokos 

17 March 2026

## Abstract

Confluence is usually read as order-independence of reduction, while flatness is usually read as path-independence of transport. This paper argues that their local cores coincide once rewriting is equipped with its missing two-dimensional cells. The relevant object is not the bare reduction graph but a reduction 2-complex whose 0-cells are states, whose 1-cells are one-step rewrites, and whose generating 2-cells are local peaks or, in structured rewriting settings, reachable critical pairs. In that setting, elementary flatness is exactly local confluence. We prove this core equivalence and record the supporting fact that confluence plus normalization yields a unique normal form from a chosen start state. We then show computationally that the corrected local/global picture survives exhaustive finite-ARS stress tests in the terminating regime, curated string-rewriting examples, and a curated left-linear term-rewriting bridge with genuine nonroot overlap. These experiments also show that bare graph-flatness surrogates fail, while reachable critical-pair defects align with the observed local obstruction layer. Finally, explicit before/after examples show that completion can be read as holonomy elimination: the local generator persists while its defect disappears. The full finite terminating left-linear TRS theorem package remains to be written in full generality, but the correct object, local generators, and evidence boundary are now fixed.

## 1 Introduction

Confluence is rewriting theory’s cleanest order-independence principle: if a state reduces in two different ways, the divergence should be joinable [1, 8, 4, 14]. Discrete transport geometry packages the same intuition in a different language: transport should be independent of route, and the obstruction is holonomy [10, 2]. This paper argues that their local cores coincide once rewriting is given its missing two-dimensional cells. The relevant object is not the bare reduction graph. It is a reduction 2-complex whose 0-cells are states, whose 1-cells are one-step rewrites, and whose generating 2-cells are local peaks or, in structured rewriting settings, reachable critical pairs [11, 7, 6].

The first temptation is also the first mistake. If one tries to read flatness directly from the bare reduction graph, one obtains surrogates that are vacuous or misleading in the terminating settings studied here. Section 3 shows this explicitly: directed-loop flatness collapses to acyclicity on the curated terminating examples, while ordinary undirected graph cycle structure is not diagnostic for confluence. The negative result is therefore constructive. It tells us where the geometric data must live: not on graph loops alone, but on elementary branching cells.

Once those missing 2-cells are restored, the local theorem-level core becomes clean. Section 5 proves that elementary flatness is equivalent to local confluence. The paper then shows computationally that the corrected local/global picture survives exhaustive finite-ARS stress tests

in the terminating regime, curated bounded string-rewriting examples, and a curated left-linear term-rewriting bridge with genuine nonroot overlap. In those structured settings, the relevant local generators are reachable critical pairs, and the before/after completion examples show that defect elimination can be read as holonomy elimination.

This paper is also a mathematics-facing contribution to the Six Birds line of work. The general closure vocabulary comes from [15], while the treatment of route mismatch as a disciplined mathematical defect was developed earlier in [16]. The present paper does not repeat that general background. Instead, it uses that vocabulary to isolate a specific claim about rewriting: the local obstructions to confluence are carried by the generating 2-cells on which elementary holonomy is tested.

The paper is organized around three levels of claim. First, there is a proved core: elementary flatness is equivalent to local confluence, and confluence plus normalization yields a unique normal form from a chosen start state. Second, there is a strong computational and structural bridge: exhaustive finite-ARS audits, curated string and term-rewriting examples, and explicit reduction 2-complex artifacts all support the corrected local geometric picture. Third, there is a deferred theorem package: the full finite terminating left-linear term-rewriting formulation in the exact target form remains to be written as a general theorem rather than a curated bridge. Section 9 makes this boundary explicit.

**Contributions.** This paper makes four contributions.

1. It identifies the correct geometric object for rewriting: a reduction 2-complex rather than a bare reduction graph.
2. It proves the local formal core that elementary flatness is exactly local confluence (Theorem 5.6), together with the supporting uniqueness-of-normal-form fact under confluence and normalization (Proposition 5.9).
3. It shows computationally that the corrected local/global picture survives exhaustive finite-ARS stress tests and curated string and left-linear term-rewriting examples, while graph-only flatness surrogates fail (Sections 3 and 7).
4. It gives an explicit operational reading of completion as holonomy elimination in before/after string and term-rewriting pairs (Section 8).

The rest of the paper proceeds as follows. Section 2 fixes the paper’s scope and its minimal Six Birds placement. Section 3 rules out the wrong graph-only formulation. Section 4 introduces the reduction 2-complex and elementary holonomy language, and Section 5 proves the theorem-level core. Section 6 identifies reachable critical pairs as the structured rewriting generators of local curvature, Section 7 records the finite-ARS, string, and term-rewriting evidence base, and Section 8 packages the before/after examples as holonomy-elimination case studies. Section 9 then states explicitly what is proved, what is strongly supported, and what remains final theorem-writing work.

## 2 Preliminaries and Scope

This section fixes the paper’s minimal vocabulary and, just as importantly, its boundary. The formal theorem layer is intentionally narrower than the eventual full theorem package suggested by the paper’s central slogan, while the computational layer is intentionally broader than a toy demonstration. Keeping those two levels distinct is essential for the paper’s claim discipline.

## 2.1 Rewriting and transport viewpoints

We work throughout with rewriting systems presented at three levels of structure. The base level is abstract rewriting: states and a one-step relation. The second level is structured rewriting, where local branchings arise from overlaps of rules and are recorded by critical pairs [9, 1, 14]. The third level is the geometric packaging introduced in Section 4, where states become 0-cells, one-step rewrites become 1-cells, and local peaks or reachable critical pairs become generating 2-cells.

At the abstract level, the central notions are reachability, joinability, local confluence, confluence, and normal form. These are defined formally in Section 5 and are standard in rewriting theory [1, 8, 4, 14]. At the structured level, the central question is how the relevant elementary branchings are generated. In the string and term-rewriting settings used here, the correct generator layer is the layer of reachable critical pairs relative to chosen starts and exploration bounds. At the geometric level, the same question is rephrased as a local path-dependence problem: does a generating branching cell admit a filler to a common witness, or does a genuine elementary route mismatch remain?

## 2.2 Six Birds placement

*Remark 2.1* (Six Birds primitives and the present paper). We use the Six Birds vocabulary of closure mechanics introduced in [15]. The six primitives are:

1. **P1**: operator rewrite,
2. **P2**: constraints or feasibility gating,
3. **P3**: route mismatch / holonomy,
4. **P4**: staging,
5. **P5**: packaging,
6. **P6**: accounting / audit.

The present paper is primarily a **P3** paper. Its central object is local route mismatch in rewriting systems. **P5** and **P6** play supporting roles by packaging the rewriting layers and by keeping the defect structure explicit in machine-readable artifacts and audits. The mathematical development below is otherwise self-contained.

Within the internal line of work, the present paper should be read as a mathematics-facing sequel to [16]. The broad finite-state and coarse-graining lineage developed in [17] is relevant background, but it is not re-explained here. The paper uses that background only to situate its scope; its main task is narrower, namely to isolate the correct local geometric object for confluence.

## 2.3 Scope and evidence boundary

The paper works in three complementary regimes.

First, it proves a core theorem at the level of abstract rewriting systems: elementary flatness is equivalent to local confluence, and confluence together with normalization yields a unique normal form from a chosen start state. This is the theorem-level core of the paper.

Second, it stress-tests the resulting local/global picture in a finite computational regime. On the finite-ARS side, the repository exhaustively checks all labeled systems up to four states in the terminating regime. On the structured rewriting side, it studies curated bounded string and

left-linear term-rewriting examples. These examples are not treated as substitutes for general theorems. They are treated as bridge evidence that the corrected local object and the corrected local generators are the right ones.

Third, it makes the two-dimensional structure explicit. The repository now contains reduction 2-complex / elementary holonomy artifacts that package the relevant 0-cells, 1-cells, generating 2-cells, boundary data, canonical fillers, and defect labels. This means that the geometric language used in the paper is operational rather than merely analogical.

Two scope restrictions are worth making explicit. The first is negative: the paper does not read flatness from the bare reduction graph, and it does not identify ordinary graph-loop structure with the local obstruction to confluence. The second is positive: the paper’s final general target is the finite terminating left-linear term-rewriting setting, but that target is not yet claimed here as a fully general theorem. What is claimed here is the proved abstract core, together with a strong computational and structural bridge that fixes the correct object and the correct local generators.

## 2.4 Guide to the remaining sections

With this scope fixed, the logic of the paper is straightforward. Section 3 rules out the wrong graph-only formulation. Section 4 introduces the corrected local geometric object. Section 5 proves the theorem-level core. Section 6 identifies reachable critical pairs as the relevant structured rewriting generators. Section 7 records the stress tests that now close the experimental side of the project, and Section 8 shows how explicit before/after rule additions can be read as holonomy elimination. The final section, Section 9, states cleanly what the manuscript has proved, what it has strongly supported, and what still remains theorem-writing work.

## 3 Why the Bare Graph Is Not Enough

The most tempting geometric reading of confluence is also the wrong one: one tries to attach “flatness” directly to the bare reduction graph and then hopes that graph-theoretic path information will recover order-independence of reduction. The computational evidence in this project shows that this move fails for a structural reason. A graph records a 1-skeleton. Confluence, however, is tested on specific local branchings, and those branchings are inherently 2-dimensional data [8, 14, 11, 6]. The reduction graph therefore remembers too little: it records paths, but not which pairs of paths are the elementary local comparisons whose agreement matters.

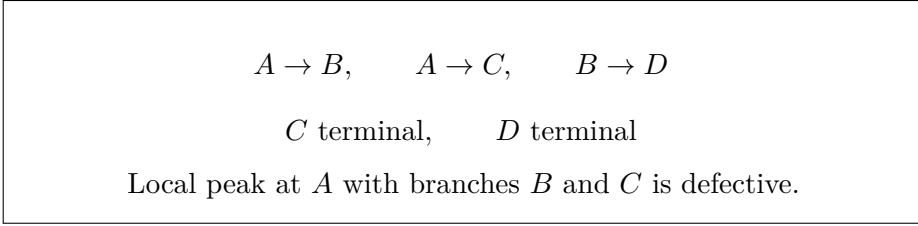
Two especially tempting graph-only surrogates were stress-tested in the finite-ARS regime.

1. A *directed-loop flatness* surrogate, which declares a system “flat” whenever the directed reduction graph has no directed cycle.
2. An *undirected cycle-rank* surrogate, which measures only the cycle-space rank of the underlying simple undirected graph.

Neither surrogate captures confluence. The first is too weak, because in terminating examples it collapses to a vacuous acyclicity test. The second is pointed at the wrong object, because ordinary graph cycles do not encode the local branching cells on which joinability is decided.

The finite-ARS audits make the failure precise.

**Proposition 3.1** (Bare graph surrogates are not diagnostic for confluence). *The graph-only surrogates above do not diagnose confluence, even in the small terminating finite-ARS regime explored in this paper.*



**Figure 1:** A nonconfluent finite ARS whose directed reduction graph is acyclic. The obstruction lives on the local branching cell at the common source, not on an ordinary graph loop, which is why graph-only flatness surrogates miss it.

**Table 1:** Curated finite-ARS comparison between graph-only surrogates and rewriting-relevant defect signals. All four examples are naively flat under the directed-loop surrogate, yet  $E002$  and  $E004$  are nonconfluent; meanwhile  $E001$  and  $E003$  are confluent despite positive undirected cycle rank.

Example	Naively flat?	Undirected cycle rank	Nonjoinability defects	Confluent?
$E001$ (diamond)	yes	1	0	yes
$E002$ (fork)	yes	0	1	no
$E003$ (delayed join)	yes	1	0	yes
$E004$ (counterexample)	yes	0	2	no

*Audit summary.* The curated finite-ARS examples already exhibit the mismatch clearly. In the examples  $E001$ – $E004$ , all four reduction graphs are terminating and therefore acyclic as directed graphs, so the naive directed-loop criterion declares all four “flat.” Yet  $E002$  and  $E004$  are nonconfluent and have nonzero local defects. Thus directed-loop flatness produces false positives.

The undirected cycle-rank surrogate fails in the opposite way. The example  $E004$  is nonconfluent while having undirected cycle rank zero, so graph-loop absence does not imply confluence. Conversely, the examples  $E001$  and  $E003$  are confluent while having positive undirected cycle rank, so graph-loop presence does not imply nonconfluence. The graph cycle structure is therefore not the right local obstruction carrier.

These curated mismatches are consistent with the exhaustive finite-ARS audit on all labeled systems up to four states in the terminating regime: the computational relationships that survive there are the rewriting-theoretic local/global ones, not the graph-only flatness surrogates. In particular, the exhaustive audit supports the local peak picture rather than a bare graph-loop picture.  $\square$

The lesson is not that path-independence language should be abandoned. The lesson is that it must be attached to the correct object. Confluence is not a statement about arbitrary loops in the 1-skeleton; it is a statement about the agreement of specific local branchings. Those branchings become visible only after the missing 2-cells are restored. This is the motivation for the reduction 2-complex introduced in Section 4.

Stated differently: the failure of the bare graph does not weaken the bridge to geometry. It sharpens it. The experiments show exactly where the geometric data must live. Elementary flatness cannot be read from the graph alone because the relevant local obstruction is not an ordinary graph loop. It is an elementary branching cell with two boundary routes and, when joinable, a filler to a common witness. That corrected local picture is what the next section makes precise.

*Remark 3.2* (From graph loops to branching cells). The negative result of this section is therefore a positive design constraint for the rest of the paper. Any viable confluence-as-flatness statement

must be formulated on a structure that remembers local branching cells. In the present paper, that structure is the reduction 2-complex. The local obstructions of interest are then carried by generating 2-cells, not by graph loops in the bare reduction graph.

## 4 Reduction 2-Complexes and Elementary Holonomy

The bare reduction graph records states and one-step rewrites, but it does not yet record the local two-route data on which confluence is tested. The guiding correction in this paper is that the relevant geometric object is therefore not the graph alone, but a *reduction 2-complex*: a directed 1-skeleton together with generating 2-cells for the local branching patterns that matter. In the finite abstract rewriting systems used earlier in the paper, those generating 2-cells are local peaks. In the string and term-rewriting examples used later, they are reachable critical pairs. The common point is that each generating 2-cell compares two admissible one-step branches from a common source.

### 4.1 From reduction graphs to reduction 2-complexes

**Definition 4.1** (Reduction 2-complex). For the purposes of this paper, a *reduction 2-complex* associated to a rewriting system  $R$  consists of the following data.

1. A set of *0-cells*, namely the states, words, or terms under consideration.
2. A set of directed *1-cells*, namely the one-step rewrites  $x \rightarrow y$ .
3. A set of generating *2-cells*, each attached to a local two-route branching pattern with common source. In the finite-ARS setting these are local peaks. In the string and term-rewriting settings used in this paper, these are reachable critical pairs relative to the chosen start states and exploration bounds.

The point of Definition 4.1 is modest but essential. We are not introducing the full apparatus of higher gauge theory here. We are isolating the minimal two-dimensional structure needed to talk about local path dependence. The reduction graph supplies the 1-skeleton; the generating peaks or reachable critical pairs supply the missing 2-cells [11, 7, 6, 12].

*Remark 4.2* (Why the bare graph is insufficient). A graph by itself can encode paths, but not which pairs of paths are the elementary local branchings whose agreement is at stake. Curvature language therefore cannot be attached to the bare reduction graph alone in any informative way. The relevant local obstruction lives on generating 2-cells, not merely on loops in the 1-skeleton. This is exactly why the graph-only flatness surrogates examined in Section 3 fail.

### 4.2 Elementary holonomy

Each generating 2-cell compares two boundary routes with the same source. If those routes can be filled to a common witness, the local discrepancy disappears. If they cannot, the local branching remembers a genuine route mismatch.

**Definition 4.3** (Boundary data and fillers). Let  $\sigma$  be a generating 2-cell with source  $x$  and one-step branches to  $y_\ell$  and  $y_r$ . Its boundary data consist of

$$x \rightarrow y_\ell \quad \text{and} \quad x \rightarrow y_r.$$

A *filler* for  $\sigma$  is a common descendant  $w$  together with directed paths

$$y_\ell \rightarrow^* w \quad \text{and} \quad y_r \rightarrow^* w.$$

When such a filler exists, we call  $w$  a *join witness* for  $\sigma$ .

**Definition 4.4** (Elementary holonomy status). Let  $\sigma$  be a generating 2-cell in a reduction 2-complex.

- $\sigma$  has *trivial elementary holonomy* if it admits a filler in the sense of Definition 4.3.
- $\sigma$  has *nontrivial elementary holonomy* if no such filler exists.

This terminology is intentionally local. The paper does not need a full group-valued holonomy theory. What it needs is a precise way to say whether the two elementary routes attached to a generating branching agree after completion to a common witness. Trivial elementary holonomy is exactly that agreement; nontrivial elementary holonomy is exactly its failure.

In the machine-readable artifacts used later in the paper, two additional pieces of information are tracked.

1. A canonical *join witness* and canonical filler paths whenever the 2-cell is joinable.
2. A canonical *defect label* whenever it is not. The primary label is `nonjoinable`. In terminating bounded examples we also record normal-form outcome mismatch as a secondary diagnostic.

These implementation choices make the local path-dependence story explicit without changing the underlying mathematical point: the first and indispensable distinction is whether the generating 2-cell is fillable at all.

*Remark 4.5* (Elementary holonomy and the theorem-level core). The theorem-level core proved in Section 5 uses only joinability of elementary peaks. The normal-form mismatch diagnostic belongs to the computational layer for terminating explored examples and is useful precisely because it refines the nontrivial cases. It is not needed for the basic equivalence between elementary flatness and local confluence.

### 4.3 Representative domains

The same two-dimensional pattern appears in the three domains used in this paper.

- In finite abstract rewriting systems, the generating 2-cells are local peaks.
- In bounded string-rewriting examples, the generating 2-cells are reachable critical pairs relative to the chosen start words.
- In the left-linear term-rewriting bridge, the generating 2-cells are reachable critical pairs relative to the chosen start terms.

Across all three domains, the reduction 2-complex records the same local question: given two admissible elementary branches from a common source, is there a canonical filler to a common witness, or does a genuine local route mismatch remain?

This is the sense in which the paper uses the phrase *elementary holonomy*. It is a local path-dependence status attached to generating 2-cells of the reduction 2-complex. Once that language is fixed, the bridge to rewriting becomes straightforward: local confluence is the statement that the relevant elementary holonomies vanish.

**Table 2:** Elementary holonomy on the representative finite-ARS, string, and term-rewriting examples used to instantiate the reduction 2-complex language. Joinable generators are trivial; defective generators are nontrivial.

Example	Domain	2-cells	Trivial	Nontrivial	Witness / defect
$E001$ (diamond)	finite ARS	1	1	0	D
$E002$ (fork)	finite ARS	1	0	1	nonjoinable
$E007$ (before)	string	1	0	1	nonjoinable
$E008$ (after)	string	1	1	0	y
$TRS003$ (nested)	term rew.	1	0	1	nonjoinable
$TRS004$ (resolved)	term rew.	1	1	0	p(a)

## 5 Core Theorem: Elementary Flatness and Local Confluence

We now state and prove the rigorous core used throughout the paper. The setting for this section is an abstract rewriting system  $(X, \rightarrow)$ , and  $\rightarrow^*$  denotes the reflexive-transitive closure of  $\rightarrow$ . The theorem is intentionally local: it identifies the exact formal content of the slogan that “flatness” is the absence of elementary route mismatch.

### 5.1 Joinability, local confluence, and elementary flatness

**Definition 5.1** (Joinability). Two states  $b, c \in X$  are *joinable* if there exists some  $d \in X$  such that

$$b \rightarrow^* d \quad \text{and} \quad c \rightarrow^* d.$$

**Definition 5.2** (Normal forms). A state  $n \in X$  is a *normal form* if there is no state  $m \in X$  such that  $n \rightarrow m$ .

**Definition 5.3** (Local confluence). The rewriting system  $(X, \rightarrow)$  is *locally confluent* if for all  $a, b, c \in X$ ,

$$a \rightarrow b \quad \text{and} \quad a \rightarrow c \quad \implies \quad b \text{ and } c \text{ are joinable.}$$

**Definition 5.4** (Elementary peaks and elementary flatness). An *elementary peak* is a triple  $(a, b, c)$  such that

$$a \rightarrow b, \quad a \rightarrow c, \quad b \neq c.$$

The rewriting system  $(X, \rightarrow)$  is *elementarily flat* if every elementary peak is joinable.

**Lemma 5.5** (Reflexive joinability). *Every state is joinable with itself.*

*Proof.* Let  $x \in X$ . Since  $\rightarrow^*$  is reflexive, we have  $x \rightarrow^* x$ . Taking  $d = x$  in Definition 5.1, we conclude that  $x$  is joinable with itself.  $\square$

**Theorem 5.6** (Elementary flatness iff local confluence). *A rewriting system is elementarily flat if and only if it is locally confluent.*

*Proof.* Assume first that  $(X, \rightarrow)$  is locally confluent. Let  $(a, b, c)$  be an elementary peak. By Definition 5.4, we have  $a \rightarrow b$  and  $a \rightarrow c$ . By local confluence,  $b$  and  $c$  are joinable. Hence every elementary peak is joinable, so the system is elementarily flat.

Conversely, assume that  $(X, \rightarrow)$  is elementarily flat. To prove local confluence, let  $a, b, c \in X$  satisfy  $a \rightarrow b$  and  $a \rightarrow c$ . There are two cases.

If  $b = c$ , then  $b$  and  $c$  are joinable by Lemma 5.5.

If  $b \neq c$ , then  $(a, b, c)$  is an elementary peak. By elementary flatness,  $b$  and  $c$  are joinable.

In both cases, the two one-step descendants of  $a$  are joinable. Therefore the system is locally confluent.  $\square$

*Remark 5.7* (What the theorem does and does not say). Theorem 5.6 is the formal core of the paper. It does not yet identify which local branchings generate the relevant 2-cells in a given rewriting formalism; that is the role of critical pairs in Section 6. Nor does it by itself supply a local-to-global theorem such as Newman-style termination plus local confluence [8, 1, 4, 14]. Its point is narrower and more precise: once the correct elementary branchings have been identified, “flatness” at that level is exactly local confluence.

## 5.2 A supporting global normal-form fact

The local theorem above is the main formal statement needed for the paper’s geometric bridge. We also record the standard supporting fact that confluence plus normalization yields a unique normal form from any chosen start state.

**Definition 5.8** (Confluence and normalization). The rewriting system  $(X, \rightarrow)$  is *confluent* if for all  $a, b, c \in X$ ,

$$a \rightarrow^* b \quad \text{and} \quad a \rightarrow^* c \quad \implies \quad b \text{ and } c \text{ are joinable.}$$

It is *normalizing* if for every  $a \in X$  there exists some normal form  $n$  such that

$$a \rightarrow^* n.$$

**Proposition 5.9** (Confluence and normalization give a unique normal form). *Assume that  $(X, \rightarrow)$  is confluent and normalizing. Then for every  $a \in X$  there exists a unique normal form  $n$  such that  $a \rightarrow^* n$ .*

*Proof.* Fix  $a \in X$ . Since the system is normalizing, there exists at least one normal form  $n$  with  $a \rightarrow^* n$ .

To prove uniqueness, suppose that  $n$  and  $m$  are both normal forms and that

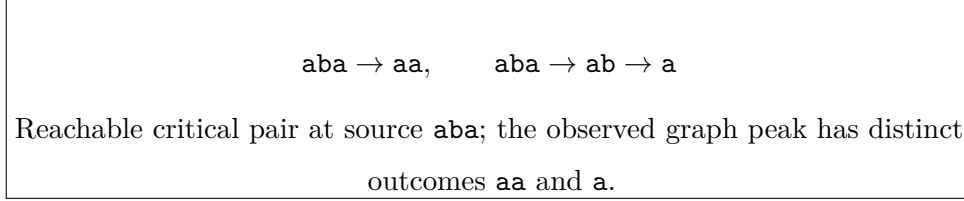
$$a \rightarrow^* n, \quad a \rightarrow^* m.$$

By confluence,  $n$  and  $m$  are joinable. Hence there exists some  $d$  such that

$$n \rightarrow^* d \quad \text{and} \quad m \rightarrow^* d.$$

But  $n$  is a normal form, so the only state reachable from  $n$  is  $n$  itself; therefore  $d = n$ . Likewise, since  $m$  is a normal form, we must have  $d = m$ . Hence  $n = m$ .  $\square$

*Remark 5.10* (Lean formal core). The definitions and baseline equivalences used in this section are mirrored in the Lean development summarized in Appendix A. In particular, the formal core contains the elementary-flatness/local-confluence equivalence together with the corresponding uniqueness-of-normal-form statement under confluence and normalization.



**Figure 2:** The string overlap example  $E006$ . The system has two system critical pairs, but only the reachable one from the designated start word appears in the explored reduction complex. Its defect matches the unique observed local peak.

## 6 Critical Pairs as Curvature Generators

Theorem 5.6 identifies the formal core of the paper at the level of abstract rewriting systems: once the relevant elementary branchings have been identified, local confluence is exactly elementary flatness. In structured rewriting formalisms, however, those elementary branchings are not presented to us as abstract triples. They arise from concrete overlaps of rewrite rules. This is the point at which critical pairs enter [9, 1, 14].

The guiding claim of this section is that critical pairs are not merely a practical confluence checklist. In the structured rewriting settings studied here, their reachable layer is the structured rewriting analogue of the generating 2-cells introduced in Section 4 [13, 12, 7, 6, 11]. Put differently: in the examples studied here, reachable critical pairs are the local generators on which elementary holonomy is tested. When a reachable critical pair is defective, the corresponding generating 2-cell has nontrivial elementary holonomy. When that defect is removed, the corresponding elementary holonomy disappears.

### 6.1 Strings: overlaps, reachability, and observed peaks

In the bounded string-rewriting examples of this paper, the local branchings of interest arise from overlaps of string rules. System-level critical pairs therefore give the natural candidate list of generating 2-cells [3]. But the correct comparison basis for the explored reduction complex is not the raw system-wide list. It is the *reachable* subset relative to the designated start words and exploration bounds.

This distinction matters already in the smallest nonconfluent overlap example. In  $E006$ , the system has two critical pairs, but only one of them is reachable from the chosen start word **aba**. The explored graph therefore displays one local peak, not two. The local obstruction seen in the explored reduction complex is not the entire system-wide overlap inventory; it is the reachable critical-pair layer. By contrast, in the sorting example  $E005$  and in the completion pair  $E007$ – $E008$ , the reachable and graph-visible local layers coincide exactly.

### 6.2 Reachable critical pairs versus graph peaks

The computational bridge developed in the repository compares these two local layers by a common signature:

(source, unordered pair of branch terms).

On the string examples, the reachable critical-pair signatures and the graph-peak signatures agree exactly.

**Table 3:** System critical pairs, reachable critical pairs, graph peaks, and reachable defect counts in the curated string and term-rewriting examples. The relevant local comparison basis for the explored reduction complex is the reachable critical-pair layer.

Example	Domain	System CPs	Reach. CPs	Graph peaks	Defective CPs	Confluent?
<i>E005</i> (sort)	string	1	1	1	0	yes
<i>E006</i> (overlap)	string	2	1	1	1	no
<i>E007</i> (before)	string	1	1	1	1	no
<i>E008</i> (after)	string	1	1	1	0	yes
<i>TRS002</i> (root)	term rew.	1	1	1	1	no
<i>TRS003</i> (nested)	term rew.	1	1	1	1	no
<i>TRS004</i> (resolved)	term rew.	1	1	1	0	yes

**Proposition 6.1** (Reachable critical pairs capture the observed local obstruction set in the curated string examples). *For the curated string examples  $E005$ – $E008$ , the reachable critical pairs match the graph-extracted local peaks, and their defect signatures agree.*

*Audit summary.* The string sorting example  $E005$  has one system critical pair, one reachable critical pair, and one graph peak; all are joinable, and all local defect counts vanish. The overlap example  $E006$  has two system critical pairs but only one reachable critical pair from the chosen start word. The explored graph correspondingly has one local peak, and that reachable critical pair is defective: both the nonjoinability and normal-form outcome-mismatch diagnostics are nonzero. The completion-before example  $E007$  has one reachable critical pair and one graph peak, both defective. The completion-after example  $E008$  has one reachable critical pair and one graph peak, both joinable. In all four cases the reachable-signature comparison is exact, and the defective-signature comparison is exact as well. Thus, on the curated string examples, reachable critical pairs capture the local obstruction set actually seen in the explored reduction complex.  $\square$

This is the sense in which reachable critical pairs behave as curvature generators in the string layer. They are the local overlap-generated 2-cells on which elementary holonomy is tested. These reachable generators are the ones that appear in the explored reduction complex for the chosen starts.

### 6.3 Left-linear term-rewriting bridge

The term-rewriting prototype pushes the same local picture beyond strings. The root-overlap example  $TRS002$  serves as a smoke test: it has a single reachable critical pair with source term  $f(a, b)$  and branch terms  $g(b)$  and  $h(a)$ , and the explored reduction graph exhibits exactly the corresponding defective local peak. More importantly, the nested-overlap pair  $TRS003$ – $TRS004$  shows that this agreement survives genuine nonroot overlap structure.

In  $TRS003$ , the two rules

$$f(g(x)) \rightarrow p(x), \quad g(a) \rightarrow b$$

produce a reachable critical pair at source term  $f(g(a))$  with branch terms  $p(a)$  and  $f(b)$ . That critical pair is defective, and the explored reduction graph is nonconfluent from the chosen start term. In  $TRS004$ , the added rule

$$f(b) \rightarrow p(a)$$

$TRS003$	$TRS004$	
$f(g(a)) \rightarrow p(a)$	$f(g(a)) \rightarrow p(a)$	Same local generator, resolved by a filler in the right-hand system.
$f(g(a)) \rightarrow f(b)$	$f(g(a)) \rightarrow f(b) \rightarrow p(a)$	

**Figure 3:** The left-linear term-rewriting bridge.  $TRS003$  and  $TRS004$  share the same reachable local generator at  $f(g(a))$  with branches  $f(b)$  and  $p(a)$ ; the added rule in  $TRS004$  supplies a filler rather than removing the generator.

makes the same local branching joinable. The critical-pair source and branch terms remain the same; what changes is the existence of a filler, and therefore the elementary holonomy status of the corresponding generating 2-cell.

**Proposition 6.2** (The curated left-linear term-rewriting bridge exhibits the same local generator pattern). *In the curated left-linear term-rewriting examples  $TRS002$ – $TRS004$ , reachable critical pairs agree with graph-visible local peaks, and their defect status agrees with the local confluence behavior observed in the explored reduction graphs.*

*Audit summary.* The root-overlap example  $TRS002$  has one reachable critical pair, one graph peak, and one nontrivial local defect. The nested-overlap example  $TRS003$  also has one reachable critical pair and one graph peak; both are defective, and the explored reduction graph is nonconfluent from the chosen start term. After the additional rule is added in  $TRS004$ , the same reachable critical-pair source and branch terms remain, but the defect disappears: the reachable critical pair becomes joinable, the graph-visible local peak becomes fillable, and the explored reduction graph becomes confluent from the chosen start term. In all three examples, the reachable-signature comparison between critical pairs and graph peaks is exact.  $\square$

The term-rewriting bridge matters because it shows that the string-level picture is not an accident of substring rewriting. Once reachable critical pairs are treated as the local overlap-generated branching cells in the explored setting, the same curvature-generator reading survives in a genuine first-order rewriting setting, at least in the curated left-linear regime implemented here.

*Remark 6.3* (Why the qualifier *reachable* matters). The phrase *critical pairs are curvature generators* would be too coarse if interpreted as a statement about every system critical pair independently of context. The correct local layer for the explored reduction complex is the reachable critical-pair layer relative to chosen starts. This is not a technicality. It is exactly what separates the system-wide overlap inventory from the local generator set that actually appears in the reduction 2-complex under study.

Taken together, the string and left-linear term-rewriting examples justify the paper’s rewriting-specific bridge claim: in structured rewriting formalisms, the relevant generating 2-cells are supplied by reachable critical pairs, and the defective ones are precisely the local generators of nontrivial elementary holonomy. The next section places this local picture inside the broader computational evidence base.

## 7 Computational Bridge and Stress Tests

The preceding sections isolate the corrected geometric object and the theorem-level local core. This section records how that picture survives the implemented stress tests. The evidence comes in three layers:

**Table 4:** Exhaustive finite-ARS audit on all labeled systems up to four states. On the terminating subset, all four discrepancy classes vanish.

Scope	Total systems	Terminating	LF≠LC	LF & ¬Conf	Conf & ¬UNF	UNF & ¬Conf
$n \leq 4$	66 066	572	0	0	0	0

1. an exhaustive audit of all labeled finite abstract rewriting systems on up to four states in the terminating regime;
2. curated bounded string-rewriting examples;
3. a curated left-linear term-rewriting bridge with genuine nonroot overlap.

The role of this section is not to replace theorem-writing. Its role is to show that once the object and the local generators are chosen correctly, the expected local/global relationships remain stable across every computational regime implemented in the repository.

## 7.1 Exhaustive finite-ARS audit

The cleanest stress test is the exhaustive finite-ARS audit. The search space contains every labeled directed rewriting relation on  $n$  states for  $n \leq 4$ , including self-loops. This yields

$$2^1 + 2^4 + 2^9 + 2^{16} = 66066$$

systems in total. Of these, 572 are terminating.

**Proposition 7.1** (Exhaustive finite-ARS stress test). *Within the terminating finite-ARS search space on up to four states, the four target discrepancy classes tracked in the repository are empty.*

*Audit summary.* The search checks 2, 16, 512, and 65536 labeled systems at sizes 1, 2, 3, and 4 respectively, for a total of 66066. The terminating counts are 1, 3, 25, and 543, for a total of 572 terminating systems. On this terminating subset, the discrepancy counts

```

local_flatness_vs_local_confluence,
terminating_local_flatness_not_confluent,
terminating_confluent_not_unique_reachable_normal_forms,
terminating_unique_reachable_normal_forms_not_confluent

```

are all zero. Equivalently, on the terminating finite-ARS regime explored here, the local peak picture survives every system in the search space while the naive graph-only flatness surrogates do not.  $\square$

The same finite-ARS infrastructure was also used outside the safe theorem hypotheses. Within the same small search bound, the boundary-counterexample search found explicit witnesses to the weakened claims obtained by dropping termination or replacing joinability-based local flatness with naive graph or normal-form surrogates. The positive finite-ARS results are therefore not an artifact of an impoverished search space; they are specific to the correct hypotheses.

**Table 5:** Curated string examples. The table reports explored graph size, reachable critical-pair counts, reachable defect counts, normal forms, and confluence status for  $E005$ – $E008$ .

Example	States	Edges	Reach. CPs	Defective CPs	Normal forms	Confluent?
$E005$	6	6	1	0	$\mathbf{abc}$	yes
$E006$	4	3	1	1	$\mathbf{a, aa}$	no
$E007$	3	2	1	1	$\mathbf{x, y}$	no
$E008$	3	3	1	0	$\mathbf{y}$	yes

**Table 6:** Curated left-linear term-rewriting bridge. The table reports explored graph size, reachable critical-pair counts, reachable defect counts, normal forms, and confluence status for  $TRS001$ – $TRS004$ .

Example	States	Edges	Reach. CPs	Defective CPs	Normal forms	Confluent?
$TRS001$	3	2	0	0	$\mathbf{s(s(z))}$	yes
$TRS002$	3	2	1	1	$\mathbf{g(b), h(a)}$	no
$TRS003$	3	2	1	1	$\mathbf{f(b), p(a)}$	no
$TRS004$	3	3	1	0	$\mathbf{p(a)}$	yes

## 7.2 Curated string systems

The string layer provides the first structured rewriting bridge. All four curated string examples explored completely under their chosen bounds, and all four exhibit the expected relation between global confluence, local peak defects, and reachable critical-pair defects.

The sorting example  $E005$  explores completely to a six-state, six-edge reduction graph with one local peak, one reachable critical pair, zero local defects, zero reachable critical-pair defects, and unique normal form  $\mathbf{abc}$ . The overlap example  $E006$  explores completely to a four-state, three-edge reduction graph with one local peak, two system critical pairs, one reachable critical pair, and normal forms  $\mathbf{a}$  and  $\mathbf{aa}$ ; its local peak defect counts and reachable critical-pair defect counts are both nonzero, and the explored system is nonconfluent. The completion-before example  $E007$  explores completely to a three-state, two-edge reduction graph with normal forms  $\mathbf{x}$  and  $\mathbf{y}$ ; it has one local peak, one reachable critical pair, nonzero local defects, nonzero reachable critical-pair defects, and nonconfluence. The completion-after example  $E008$  explores completely to a three-state, three-edge reduction graph with unique normal form  $\mathbf{y}$ ; it has one local peak, one reachable critical pair, zero local defects, zero reachable critical-pair defects, and confluence.

Taken together, the string examples do not merely show that some nonconfluent systems have bad critical pairs. They show the tighter statement needed for the paper: on the curated examples, the reachable critical-pair defect layer and the observed graph-peak defect layer agree exactly, and both align with the global confluence behavior of the explored system.

## 7.3 Curated left-linear term-rewriting bridge

The left-linear term-rewriting prototype extends the same local/global pattern to a genuine first-order setting. All four curated examples explore completely under their chosen bounds.

The Peano-addition smoke test  $TRS001$  explores completely to a three-state, two-edge reduction graph with unique normal form  $\mathbf{s(s(z))}$  and no critical pairs. The root-overlap example  $TRS002$  explores completely to a three-state, two-edge reduction graph with one reachable critical pair, one graph peak, normal forms  $\mathbf{g(b)}$  and  $\mathbf{h(a)}$ , and a nontrivial reachable defect. The nested-overlap

nonconfluent example *TRS003* explores completely to a three-state, two-edge reduction graph with one reachable critical pair, one graph peak, normal forms  $f(b)$  and  $p(a)$ , and a nontrivial reachable defect. The nested-overlap resolved example *TRS004* explores completely to a three-state, three-edge reduction graph with one reachable critical pair, one graph peak, unique normal form  $p(a)$ , and zero reachable defects.

The term-rewriting bridge therefore reproduces the same operational pattern seen in the string layer. The local generator persists, but its defect status changes. In the defective case the explored system is nonconfluent from the chosen start term; in the resolved case the explored system is confluent. Moreover, the reachability cross-check between critical-pair signatures and graph-peak signatures is exact on all curated TRS examples.

*Remark 7.2* (What the computational bridge does and does not establish). The computational evidence in this section is strong enough to fix the correct local geometric picture. It is not, by itself, the final theorem package for finite terminating left-linear term rewriting systems in full generality. What it does establish is that the corrected local/global story survives every implemented stress test: exhaustive finite ARS search in the terminating regime, curated string overlaps and completion pairs, and curated left-linear term-rewriting examples with genuine nonroot overlap.

## 8 Completion as Holonomy Elimination

The preceding sections identify the local generators and show that their defect status aligns with observed confluence behavior. This section packages that local picture into the paper’s most operational reading: in explicit before/after examples, a hand-specified completion step removes a local obstruction, and the corresponding elementary holonomy disappears. In this sense, completion can be read as *holonomy elimination*.

The point is deliberately modest. The paper is not claiming an algorithmic completion theorem in geometric language [9]. It is claiming that, in explicit finite examples, the addition of the right rule changes a previously nontrivial generating 2-cell into a trivial one. The local defect disappears, and the global behavior changes with it.

### 8.1 String before/after pair

The string pair *E007–E008* is the cleanest instance of this pattern. Before the added rule, the system has the two rules

$$ab \rightarrow x, \quad ab \rightarrow y.$$

From the chosen start word  $ab$ , the explored reduction graph has one local peak, one reachable critical pair, and two normal forms, namely  $x$  and  $y$ . The corresponding generating 2-cell has nontrivial elementary holonomy: there is no filler to a common witness, the local peak is defective, and the explored system is nonconfluent.

After the additional rule

$$x \rightarrow y$$

is added, the local branching source and branch words remain exactly the same. The generating 2-cell is not removed; it is *resolved*. The branch  $x$  now reduces to  $y$ , so the local branching admits a filler with join witness  $y$ . The elementary holonomy of the corresponding 2-cell becomes trivial, the local defect counts drop to zero, and the explored system becomes confluent with unique normal form  $y$ .

Before	After	
$ab \rightarrow x$	$ab \rightarrow x \rightarrow y$	The generator persists while the filler $x \rightarrow y$ becomes available.
$ab \rightarrow y$	$ab \rightarrow y$	

**Figure 4:** Before/after string completion pair. The generator at  $ab$  with branches  $x$  and  $y$  is nontrivial before the added rule and trivial after it, with join witness  $y$  in the resolved case.

Before	After	
$f(g(a)) \rightarrow p(a)$	$f(g(a)) \rightarrow p(a)$	The generator remains, but the added rule $f(b) \rightarrow p(a)$ supplies a filler.
$f(g(a)) \rightarrow f(b)$	$f(g(a)) \rightarrow f(b) \rightarrow p(a)$	

**Figure 5:** Before/after term-rewriting completion pair. The generator at  $f(g(a))$  with branches  $f(b)$  and  $p(a)$  is nontrivial before the added rule and trivial after it, with join witness  $p(a)$  in the resolved case.

## 8.2 Term-rewriting before/after pair

The same operational pattern appears in the left-linear term-rewriting bridge. In  $TRS003$ , the nested overlap between

$$f(g(x)) \rightarrow p(x) \quad \text{and} \quad g(a) \rightarrow b$$

produces a reachable critical pair with source term  $f(g(a))$  and branch terms  $p(a)$  and  $f(b)$ . In the chosen explored closure, this local branching is defective: there is no filler to a common witness, the associated generating 2-cell has nontrivial elementary holonomy, and the explored system is nonconfluent from the chosen start term.

In  $TRS004$ , the additional rule

$$f(b) \rightarrow p(a)$$

resolves the same local branching. The critical-pair source and branch terms remain the same as in  $TRS003$ , but the branch  $f(b)$  now reduces to  $p(a)$ . The corresponding generating 2-cell therefore acquires a filler with join witness  $p(a)$ . Its elementary holonomy becomes trivial, its defect disappears, and the explored system becomes confluent from the chosen start term.

## 8.3 What is eliminated

The key point in both pairs is that completion does not eliminate the local generator itself. The branching source remains, and the local comparison remains. What changes is the existence of a filler. In the language of this paper, the generating 2-cell remains present while its elementary holonomy changes from nontrivial to trivial.

This is exactly why the phrase *holonomy elimination* is the right operational reading. The local generator survives as a 2-cell in the reduction complex, but the route mismatch disappears. Measured in the computational artifacts, the count of generating cells does not have to drop; the count of *defective* generating cells drops to zero.

**Proposition 8.1** (Completion as holonomy elimination in the explicit before/after pairs). *In the explicit string pair E007–E008 and the explicit term-rewriting pair TRS003–TRS004, the added rule leaves the local generator intact while eliminating its defect. In both cases:*

1. the relevant local branching source persists;

**Table 7:** Before/after deltas for the explicit completion pairs. In both the string and term-rewriting cases, the generating cell persists while its defect count drops to zero and confluence is restored.

Pair	Before	After	Added rule	Defective gen.		Holonomy		Confluent?	
				Bef.	Aft.	Bef.	Aft.	Bef.	Aft.
string	E007	E008	$x \rightarrow y$	1	0	nontriv.	triv.	no	yes
term rew.	TRS003	TRS004	$f(b) \rightarrow p(a)$	1	0	nontriv.	triv.	no	yes

2. the generating 2-cell changes from nontrivial elementary holonomy to trivial elementary holonomy;
3. the local defect counts drop to zero;
4. the explored system changes from nonconfluent to confluent.

*Audit summary.* For the string pair, the source word  $\mathbf{ab}$  and the branch words  $\mathbf{x}$  and  $\mathbf{y}$  occur both before and after the added rule. The before-system has one reachable critical pair, one defective local peak, one nontrivial generating 2-cell, and nonconfluence. The after-system has the same reachable critical-pair source and branch words, but the branch  $\mathbf{x}$  now reaches  $\mathbf{y}$ . The corresponding generating 2-cell is therefore fillable, its elementary holonomy becomes trivial, the local defect counts vanish, and the system is confluent.

For the term-rewriting pair, the source term  $f(g(a))$  and the branch terms  $p(a)$  and  $f(b)$  likewise occur both before and after the added rule. In  $TRS003$ , the reachable critical pair is defective and the corresponding generating 2-cell has nontrivial elementary holonomy. In  $TRS004$ , the additional rule  $f(b) \rightarrow p(a)$  supplies a filler with witness  $p(a)$ . The local defect vanishes, the elementary holonomy becomes trivial, and the explored system becomes confluent from the chosen start term. Thus the local generator persists while its defect is eliminated in both before/after pairs.  $\square$

*Remark 8.2* (Boundary of the claim). Nothing in this section is meant to identify completion in full generality with a geometric algorithm on reduction complexes. The claim is narrower. In the explicit case studies implemented here, the added rule removes a local route mismatch carried by a generating 2-cell, and that removal coincides with restoration of confluence in the explored system. This is enough to justify the operational reading of completion as holonomy elimination, and no stronger algorithmic statement is needed for the paper.

## 9 What Is Proved, What Is Supported, What Remains

The paper is intentionally organized around three levels of claim. This is not a hedge. It is the correct way to separate what has already been established from what has been computationally fixed and from what still remains to be written as the final theorem package. The resulting boundary is one of the paper’s main strengths: it makes the bridge precise without pretending that every desirable general statement has already been formalized.

### 9.1 What is already proved

The theorem-level core of the paper is already in place. At the level of abstract rewriting systems, Theorem 5.6 proves that elementary flatness is equivalent to local confluence. This gives the paper

**Table 8:** Claim-support matrix for the paper. The table distinguishes statements already proved, statements strongly supported by the computational and structural evidence, and the final theorem package deferred to later work.

Claim family	Evidence type	Status
Elementary flatness $\leftrightarrow$ local confluence	Lean formal core + paper proof	proved
Confluence + normalization $\Rightarrow$ unique normal form from a start	Lean formal core + paper proof	proved
Bare graph flatness is not diagnostic for confluence	exhaustive computational + curated finite ARS	emp. supported
Reachable CP defects capture observed local obstruction set in curated strings	curated term level + structural artifact	emp. supported
Reachable CP defects capture observed local obstruction set in curated left-linear TRSs	curated term level + structural artifact	emp. supported
Completion acts as holonomy elimination in explicit before/after pairs	curated term level + structural artifact	emp. supported
Full finite terminating left-linear TRS confluence-as-flatness theorem	theorem package not yet written	deferred

its central formal sentence: once the relevant elementary branchings have been identified, local confluence is exactly the vanishing of elementary route mismatch. In the terminology of the paper, local confluence is exactly elementary flatness.

A second supporting fact is also already proved: under confluence and normalization, each chosen start state has a unique reachable normal form (Proposition 5.9). This is not the paper’s headline theorem, but it is an important supporting bridge between the local path-filling language and the usual outcome language of rewriting theory.

These core results are not merely informal manuscript claims. They are mirrored in the Lean development summarized in Appendix A. The formal core currently includes the elementary-flatness/local-confluence equivalence and the corresponding uniqueness-of-normal-form statement under confluence and normalization. The Lean layer therefore already secures the paper’s most basic theorem-level commitments.

## 9.2 What is strongly supported by the computational and structural evidence

The next layer of the paper is not theorem-writing in the strict sense, but it is substantially stronger than anecdotal experimentation. The repository now contains exhaustive, curated, and structural evidence that all points in the same direction.

First, the bare-graph formulation has been falsified in exactly the way the corrected theory predicts. Directed-loop flatness on the reduction graph is vacuous in the curated terminating examples, and undirected graph cycle structure is not diagnostic for confluence. The graph-only surrogates therefore fail not accidentally, but because they are attached to the wrong object.

Second, the finite-ARS stress tests are now strong enough to stabilize the local/global picture in the small terminating regime. The exhaustive audit over all labeled systems up to four states finds no counterexample to the expected finite-ARS relationships on the terminating subset, while the boundary-counterexample search finds explicit failures as soon as the safe hypotheses are weakened in the wrong way. The evidence therefore distinguishes sharply between the correct local peak picture and several tempting but incorrect substitutes.

Third, the curated string and left-linear term-rewriting examples support the rewriting-specific bridge. In the string layer, reachable critical-pair defects agree with the observed graph-peak defect layer and align with global confluence behavior on the explored systems. In the left-linear term-rewriting bridge, the same pattern survives genuine nonroot overlap structure. These results do not yet amount to the final fully general theorem for finite terminating left-linear term rewriting systems, but they do isolate the correct local generators and the correct comparison basis.

Fourth, the reduction 2-complex / elementary holonomy layer is now explicit. The repository contains machine-readable artifacts with 0-cells, 1-cells, and generating 2-cells; canonical join witnesses where they exist; and explicit trivial versus nontrivial elementary holonomy status. This matters because it turns the paper’s geometric language into something operational rather than rhetorical. The corrected local object is no longer just described; it is instantiated across representative finite-ARS, string, and term-rewriting examples.

Finally, the completion case studies support the paper’s most memorable operational reading. In both the string and term-rewriting before/after pairs, the relevant local generator survives while its defect is eliminated. The source of the local comparison persists, but the filler becomes available. That is exactly the sense in which the repository supports reading completion as holonomy elimination.

### 9.3 What remains theorem-writing work

What remains is no longer an experimental gap. It is theorem packaging and statement refinement.

The first remaining task is the full general theorem package for finite terminating left-linear term rewriting systems in the target form suggested by the paper: a confluence-as-flatness statement on the reduction 2-complex with critical pairs as the local curvature generators. The repository now fixes the right local picture, the right object, and the right examples, but the final general theorem still has to be written at that level of abstraction.

The second remaining task is to connect the present formulation more explicitly to the standard critical-pair completeness bridge in the full TRS setting. The curated term-rewriting examples already show that the bridge is pointing in the right direction, but the full paper theorem should make the relation to standard critical-pair methods mathematically explicit rather than merely computationally persuasive [9, 1, 14, 13, 5].

The third remaining task is presentational rather than conceptual: the reduction 2-complex language should be related, in the final theorem-level packaging, to the appropriate higher-dimensional rewriting or discrete transport vocabulary without overstating what has been proved here [11, 7, 6, 10, 2]. That is a matter of theorem phrasing and literature integration, not more implementation.

*Remark 9.1* (The remaining gap is not more code). At this point, the repository has already done the experimental work needed for this paper. The finite-ARS audits are frozen, the string and term-rewriting bridges are frozen, the reduction 2-complex layer is explicit, the completion case studies are explicit, and the claim-support matrix records the current evidence boundary. What remains is theorem writing, literature positioning, and final manuscript discipline rather than new implementation.

The practical consequence is simple. The paper can now be written honestly in three voices at once: a proved core, a strong computational bridge, and a clearly delimited deferred theorem package. That is not a weakness of the narrative. It is what makes the narrative trustworthy.

## 10 Conclusion

The paper began from a tempting but incorrect slogan: perhaps confluence can be read as flatness on the bare reduction graph. The repository evidence shows that this is the wrong level of structure. The reduction graph by itself does not remember the elementary local branchings whose agreement confluence tests. The correct object is a reduction 2-complex whose generating 2-cells are local peaks in the finite-ARS setting and reachable critical pairs in the structured rewriting settings studied here.

Once that correction is made, the local picture becomes clean. Elementary flatness is exactly local confluence. Defective local generators—concretely, reachable critical pairs in the structured settings studied here—carry nontrivial elementary holonomy. Completion, in explicit before/after examples, acts as elimination of that elementary holonomy: the generator remains, but the filler becomes available, the defect disappears, and confluence is restored.

The repository now supports this picture at three levels. The formal core is proved and mirrored in Lean. The finite-ARS regime has been stress-tested exhaustively in the terminating case. The string and left-linear term-rewriting bridges show that reachable critical-pair defects match the observed local obstruction layer in the explored reduction complex, while the reduction 2-complex artifacts make the two-dimensional structure explicit.

The final full theorem package for finite terminating left-linear term rewriting systems remains to be written in that exact generality. But the experimental and structural side of the project is now closed. The correct local object has been identified, the wrong graph-only surrogates have been ruled out, the local generators have been isolated, and the operational meaning of defect elimination has been made explicit.

The paper’s central message can therefore be stated without exaggeration. Order-independence of reduction is path-independence of transport, but only after rewriting is given its missing 2-dimensional cells. In that corrected setting, local peaks and, in the structured settings studied here, reachable critical pairs are the local generators on which curvature lives.

## A Lean Formal Core

The repository contains a small Lean development whose purpose is to secure the paper’s formal core, not to formalize the entire computational bridge. The Lean package is rooted at `ConfluenceFlat` and currently consists of two substantive files: `ConfluenceFlat/Core.lean` and `ConfluenceFlat/UniqueNormalForm.lean`.

### A.1 Formalized definitions

The file `ConfluenceFlat/Core.lean` formalizes the abstract rewriting notions used in Section 5. These include:

- finite abstract rewriting systems,
- reflexive-transitive reachability,
- joinability,
- normal forms,
- local confluence,

- elementary peaks,
- elementary flatness.

The file `ConfluenceFlat/UniqueNormalForm.lean` adds the supporting global notions used later in Section 5, including:

- confluence,
- normalization,
- uniqueness of normal form from a chosen start state,
- existence and uniqueness of normal form under confluence and normalization.

## A.2 Formalized theorem layer

The formal core currently proves the following results by name.

- In `ConfluenceFlat/Core.lean`:
  - `ConfluenceFlat.FiniteARS.joinable_refl`,
  - `ConfluenceFlat.FiniteARS.elementaryFlat_of_localConfluent`,
  - `ConfluenceFlat.FiniteARS.localConfluent_of_elementaryFlat`,
  - `ConfluenceFlat.FiniteARS.elementaryFlat_iff_localConfluent`.
- In `ConfluenceFlat/UniqueNormalForm.lean`:
  - `ConfluenceFlat.FiniteARS.reachable_eq_of_normalForm`,
  - `ConfluenceFlat.FiniteARS.eq_of_joinable_normalForms`,
  - `ConfluenceFlat.FiniteARS.uniqueNormalFormFrom_of_confluent`,
  - `ConfluenceFlat.FiniteARS.existsUniqueNormalFormFrom_of_confluent_normalizing`.

## A.3 Correspondence to the main text

These Lean theorems mirror the theorem-level commitments made in the main text. The equivalence

$$\text{elementary flatness} \iff \text{local confluence}$$

is the formal counterpart of Theorem 5.6. The existence-and-uniqueness result under confluence and normalization is the formal counterpart of Proposition 5.9. The Lean layer therefore secures the paper’s proved core while leaving the computational bridge, artifact layer, and full deferred TRS theorem package outside the scope of formalization.

## B Computational Protocol

This appendix summarizes the computational protocol used to generate the evidence layers cited throughout the paper. The purpose of the protocol is reproducibility and claim discipline rather than algorithmic novelty. The implementation repository for the experiments and code is <https://github.com/ioannist/six-birds-confluence>.

## B.1 Finite-ARS exhaustive protocol

The finite-ARS audit enumerates every labeled directed rewriting relation on  $n$  states for  $n \leq 4$ , including self-loops. This yields a total search space of

$$2^1 + 2^4 + 2^9 + 2^{16} = 66066$$

systems. Terminating systems are identified by directed-acyclicity of the finite reduction graph. For each terminating system, the repository computes local peak counts, nonjoinability defect counts, normal-form mismatch counts, local confluence, confluence, and uniqueness of reachable normal forms. The resulting summary and row-level artifacts are stored under `results/audits/`.

## B.2 Bounded string-rewriting protocol

The string-rewriting examples are explored from designated start words under explicit depth, state-count, and word-length bounds. The exploration records a completeness flag so that the paper can distinguish complete explored closures from genuinely truncated ones. In the curated examples used here, the chosen bounds are sufficient for complete exploration. The repository computes graph-level local peaks, system critical pairs, reachable critical pairs relative to the designated starts, and the corresponding defect counts and signature comparisons.

## B.3 Bounded term-rewriting protocol

The term-rewriting bridge uses a bounded first-order rewriting prototype with explicit start terms and exploration bounds. The critical-pair generator supports the left-linear overlap regime needed for the curated examples, including nonroot nonvariable-position overlaps. As in the string layer, the repository compares reachable critical-pair signatures with graph-peak signatures in the explored reduction graph and records whether the comparison is exact.

## B.4 Deterministic witness and filler selection

Across the plaquette, critical-pair, and two-complex layers, witnesses and filler paths are selected deterministically. The policy is: minimize total branch-to-witness distance, break ties lexicographically on the witness, and then choose lexicographically smallest shortest paths from each branch to the chosen witness. This policy ensures that the machine-readable artifacts are stable across runs and that the resulting local generators can be compared reliably across domains.

## B.5 Regression and artifact generation

The repository includes a regression harness that runs the unit tests, the curated finite-ARS checks, the curated structured rewriting checks, the small exhaustive audit, the diagram-generation step, and the results-ledger refresh. The frozen evidence boundary used in this paper is recorded by the final experiment summary, the claim-support matrix, and the central results ledger under `results/index.json` and `results/summary.csv`.

# C Artifact Guide

The paper is backed by a machine-readable artifact layer. This appendix gives a compact guide to that layer and to the stable entry points used in the manuscript.

## C.1 Artifact families

The major artifact families are:

- audit artifacts under `results/audits/`,
- peak artifacts under `results/peaks/`,
- plaquette artifacts under `results/plaquettes/`,
- string-rewriting artifacts under `results/string_rewrite/`,
- critical-pair artifacts under `results/critical_pairs/`,
- term-rewriting artifacts under `results/term_rewrite/`,
- completion case-study artifacts under `results/case_studies/`,
- diagram assets under `results/diagrams/`,
- reduction 2-complex / elementary holonomy artifacts under `results/two_complex/`,
- freeze artifacts under `results/freeze/`.

## C.2 Section-to-artifact map

The manuscript uses these artifact families in a disciplined way. The bare-graph falsification section relies on the naive graph audit and finite-ARS artifacts. The reduction 2-complex section relies on the two-complex artifacts. The critical-pair section relies on the critical-pair, string, and term-rewriting artifacts. The completion section relies on the case-study and two-complex layers. The final boundary-setting section relies on the freeze summary and the claim-support matrix. This section-to-artifact mapping is also recorded in machine-readable form under `paper/notes/section_map.yaml`.

## C.3 Figure and table source map

The kept figure and table slots are backed by specific artifact families rather than ad hoc manuscript data. Their intended source mapping is recorded in `paper/notes/figure_table_plan.yaml`. This makes it possible to reconstruct the provenance of every figure and table used in the paper without re-running the full implementation workflow manually.

## C.4 Using the results ledger

The stable entry points for the artifact layer are `results/index.json` and `results/summary.csv`. These files index the non-ignored result artifacts, record stable artifact identifiers, and expose the major logical collections used throughout the project. They are the recommended first stop for anyone tracing a statement in the manuscript back to its computational or structural evidence.

## References

- [1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

- [2] John C. Baez and Javier P. Muniain. *Gauge Fields, Knots and Gravity*, volume 4 of *Series on Knots and Everything*. World Scientific Publishing Company, River Edge, NJ, 1994.
- [3] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. String diagram rewrite theory III: Confluence with and without frobenius. *Mathematical Structures in Computer Science*, 32(7):829–869, 2022.
- [4] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 243–320. Elsevier, 1990.
- [5] Bernhard Gramlich and Salvador Lucas. Generalizing newman’s lemma for left-linear rewrite systems. In Frank Pfenning, editor, *Term Rewriting and Applications*, volume 4098 of *Lecture Notes in Computer Science*, pages 66–80, Berlin, Heidelberg, 2006. Springer.
- [6] Yves Guiraud and Philippe Malbos. Higher-dimensional normalisation strategies for acyclicity. *Advances in Mathematics*, 231(3–4):2294–2351, 2012.
- [7] Yves Guiraud and Philippe Malbos. Polygraphs of finite derivation type. *Mathematical Structures in Computer Science*, 28(2):155–201, 2018.
- [8] Gérard P. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.
- [9] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon, Oxford, 1970.
- [10] Claudio Meneses. Thin homotopy and the holonomy approach to gauge theories. In Fernando Galaz-García, Cecilia González-Tokman, and Juan Carlos Pardo Millán, editors, *Mexican Mathematicians in the World: Trends and Recent Contributions*, volume 775 of *Contemporary Mathematics*, pages 233–253. American Mathematical Society, 2021.
- [11] Samuel Mimram. Categorical coherence from term rewriting systems. In Marco Gaboardi and Femke van Raamsdonk, editors, *8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023)*, volume 260 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [12] Craig C. Squier. Word problems and a homological finiteness condition for monoids. *Journal of Pure and Applied Algebra*, 49(1–2):201–217, 1987.
- [13] Craig C. Squier and Friedrich Otto. The word problem for finitely presented monoids and finite canonical rewriting systems. In Pierre Lescanne, editor, *Rewriting Techniques and Applications*, volume 256 of *Lecture Notes in Computer Science*, pages 74–82, Berlin, Heidelberg, 1987. Springer.
- [14] Terese. *Term Rewriting Systems*. Number 55 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.
- [15] Ioannis Tsiokos. Six birds: Foundations of emergence calculus, 2026. Manuscript. DOI: <https://doi.org/10.48550/arXiv.2602.00134>.

- [16] Ioannis Tsiokos. To count a stone with six birds: A mathematics is a theory, 2026. Manuscript. DOI: <https://doi.org/10.5281/zenodo.18402004>.
- [17] Ioannis Tsiokos. To lay a stone with six birds: Finite-state semantics for packaging, directionality, and coarse-graining, 2026. Manuscript. DOI: <https://doi.org/10.20944/preprints202602.1699.v1>.