

Scoped Structural Rules in Combinatory Reduction

Flyxion
Independent Researcher

October 2026

Abstract

Combinatory logic turns the structural rules of logic into ordinary reduction rules: **W** duplicates an argument, **K** discards one, and **C** exchanges two. This essay studies reduction systems in which each of these combinators may act at an occurrence only when the bracketed scopes enclosing that occurrence jointly permit it. Five groups of results are established, listed in the order of the text. First, the availability of a structural rule in a system is separated from its admissibility at a particular redex, and the independence of contraction and weakening from the remaining combinators is proved by invariants of the lambda translation. Second, reduction that never moves terms across scope boundaries is confluent for every assignment of permissions, while the addition of a rule by which an argument enters a scope yields a confluent system exactly when no scope restricts anything. Absorption can be restricted: for an abstract base relation of a stated spine-based form, absorbing only arguments that are normal for the base relation does not preserve confluence, by an explicit counterexample in which a pending absorption inside the argument creates a redex that is then evaluated under the wrong permissions, while absorbing only arguments that are normal for the combined relation, or normal arguments containing no permission-sensitive constant, does preserve it. Third, in scoped term graphs the equivalence of copying and sharing holds under a coherence condition on the permissions seen by shared nodes; the condition is sufficient and not necessary, and an explicit example shows that once it fails the tree system and the two graph policies, one admitting a step when some access path permits it and one when every access path does, produce three different sets of normal forms. Fourth, admissibility cannot grow along a reduction: the effective permission of every surviving constant occurrence descends, so the scope structure of the initial term bounds the structural fragment that is ever realized without determining it. Fifth, in a more general account of boundary-governed transition systems, erasing boundaries preserves every admissible reduction, fails to reflect admissibility exactly when some scope restricts the root permission, and fails to lift reductions even when no permission is restricted, because erasure can expose redexes that the boundaries interrupt; loss of permission and loss of exposure are independent obstructions. In scoped term graphs, core steps under the universal policy, path-local absorption, and in-place absorption authorized along every access path are sound with respect to the tree relation, and a counterexample shows that each condition cannot be removed from the statement. Weaker semantic conditions for absorption and the sheaf-theoretic reading of the local-to-global question are left as proposals.

1 Introduction

The structural rules of a logic say what may be done with assumptions without regard to what the assumptions say: an assumption may be used twice, may be left unused, and may be

reordered relative to its neighbors. Combinatory logic makes these rules explicit by assigning each of them a combinator, so that the question of which structural rules a system has becomes the question of which reduction rules it contains. Schönfinkel’s original basis already separates the three tasks (Schönfinkel, 1924). Concatenative languages make the same separation visible in their surface vocabulary, where the words that copy, discard, and exchange stack items are the language’s contraction, weakening, and exchange (Pestov et al., 2010). In none of these settings does the permission to copy depend on where in a term the copying occurs.

The systems studied here add exactly that dependence. A term may contain bracketed subterms, each bracket carries a label, and each label carries a set of structural combinators that are permitted to act inside it. A redex is admissible when every enclosing bracket permits the structural combinator at its head. The brackets are therefore boundaries of structural permission. They are not boundaries of evaluation in the sense that a calculus might reserve evaluation for an innermost frontier, and nothing below assumes that the two kinds of boundary coincide; the development keeps the question of which transformations are authorized separate from the question of where evaluation takes place. The calculus is an abstraction of the setting of the circle calculus, and the essay *Duplication and Universality in the Circle Calculus* serves as its technical predecessor: that essay establishes what the combinatory machinery can express, and the present one asks what happens when those capacities become conditional on structure. Universality and admissibility are different questions, and the answers here do not depend on any result of the earlier essay beyond the definitions stated below. Results attributed by citation are inherited from the literature; the permission structures, the effective-permission calculus, the coherence condition for graphs, the restricted-absorption results, and the descent and separation results are introduced here.

Every numbered statement carries one of three tags. A statement tagged status E is established by a complete proof in the text. A statement tagged status C is proved from a cited theorem, and the verification of that theorem’s hypotheses is part of the proof. A statement tagged status P is a proposal, a conjecture, or a program, and no proof is claimed. Uniqueness of normal forms, joinability, and confluence are distinguished from determinism throughout: a system is deterministic when every term has at most one next step, which none of the systems here is, and confluence asserts only that different choices can be reconciled.

Two distinctions organize the material. The first is between a rule being available in a system and being admissible at a given redex, which Section 3 makes precise. The second is between a static classification, computed from the scope tree of a term, and a dynamic one, computed over everything the term can reduce to. Section 2 collects the unscoped background, including the exact relation between the combinator *W* and logical contraction and the proofs that contraction and weakening cannot be recovered from the remaining combinators. Section 4 proves confluence for the core scoped system, characterizes when the addition of a boundary-crossing rule preserves it, and in a final subsection analyzes restricted absorption for an abstract base relation. The graph treatment of Section 5 is extended to absorption in a final subsection there. Section 5 treats the comparison of copying with sharing in term graphs, and Section 6 proves the descent of admissibility under reduction. Section 7 gives the more abstract setting and separates loss of permission from loss of exposure, and Section 8 lists open problems.

2 Structural Combinators and Their Logical Reading

The combinators in play are I , B , C , K , W , and, for comparison, S , with reduction rules

$$\begin{array}{lll} Ix \rightarrow x, & Bfgx \rightarrow f(gx), & Cfxy \rightarrow fyx, \\ Kxy \rightarrow x, & Wfx \rightarrow fxx, & Sfgx \rightarrow fx(gx). \end{array}$$

The combinators I and B are neutral with respect to structure: they neither copy, discard, nor reorder, and they supply identity and composition. The remaining three realize the structural rules directly. The combinator C is exchange, K is weakening, and W is contraction in the literal sense that its type $(A \rightarrow A \rightarrow B) \rightarrow A \rightarrow B$ is the rule that merges two occurrences of a hypothesis. The combinator S is contraction only operationally, because it duplicates the use of its third argument while also applying two functions. The following proposition records the exact relation.

Proposition 2.1 (status E). *Over the base $\{I, B, C\}$ the combinators S and W define one another: $W = CSI$ and $S = B(BW)(BBC)$.*

Proof. For the first identity, $CSIxy \rightarrow Sxly$ by the rule of C with $f := S$, $x := I$, $y := x$; then $Sxly \rightarrow xy(Iy)$ by the rule of S , and $xy(Iy) \rightarrow xyy$ by the rule of I . For the second, write $M = B(BW)(BBC)$. The rule of B with $f := BW$, $g := BBC$, $x := f$ gives $Mfgx \rightarrow BW(BBCf)gx$. The rule of B again, with $f := W$, $g := BBCf$, $x := g$, gives $W(BBCfg)x$. The rule of W gives $BBCfgxx$. The rule of B with $f := B$, $g := C$, $x := f$ rewrites the prefix $BBCf$ to $B(Cf)$, so the term becomes $B(Cf)gx$. The rule of B with $f := Cf$, $g := g$, $x := x$ gives $Cf(gx)$, and the rule of C gives $fx(gx)$. $\square$

The proposition says that the identification of S with contraction is correct in the presence of exchange and incorrect without it. For this reason the permission structure below ranges over $\{W, K, C\}$ and treats S as a derived combinator whose admissibility is computed from those of its components.

To separate the three structural combinators from the neutral ones, translate combinators into the lambda calculus by $I^\circ = \lambda x.x$, $B^\circ = \lambda fgx.f(gx)$, $C^\circ = \lambda fxy.fyx$, $K^\circ = \lambda xy.x$, $W^\circ = \lambda fx.fxx$, and $S^\circ = \lambda fgx.fx(gx)$. A combinator reduction $T \rightarrow T'$ yields $T^\circ \rightarrow_\beta^* T'^\circ$. Call a lambda term *affine* when no variable occurs free more than once in it, no bound variable occurs more than once in the scope of its binder, and the two parts of every application have disjoint free variables. Call a lambda term a λI -term when every abstraction $\lambda x.M$ has x free in M . Both classes are closed under β -reduction, affine reduction strictly decreases size, and β -reduction in λI preserves the set of free variables; these are classical facts: the preservation of free variables under β -reduction of λI -terms, the termination of affine reduction, and the Church–Rosser theorem are all treated in Barendregt (1984) and, for λI , in Church (1941).

Proposition 2.2 (Separation; status C). *(a) No term over $\{I, B, K, C\}$ defines W or S . (b) No term over $\{I, B, C, W, S\}$ defines K .*

Proof. (a) The translations of I, B, K, C are closed affine terms, so the translation T° of any combination is a closed affine term. Suppose $Tfx \rightarrow_\beta fxx$. Then $T^\circ fx =_\beta fxx$. The term $T^\circ fx$ is affine, so it has a normal form in which x occurs at most once, while fxx is itself a normal form in which x occurs twice. By the Church–Rosser theorem the two normal forms coincide, a contradiction. The argument for S is identical with $fx(gx)$. (b) The translations

of the listed combinators are λ I-terms. If $T a b \rightarrow a$ then $T^\circ a b =_\beta a$, hence $T^\circ a b \rightarrow_\beta^* a$ by Church-Rosser. The free variables of $T^\circ a b$ are $\{a, b\}$ and are preserved along λ I reduction, but those of a are $\{a\}$, a contradiction. $\square$

The independence of exchange from $\{I, B, K, W\}$ is the delicate member of the triple, because deletion and duplication can disturb the left-to-right order of variable occurrences in ways that a naive order invariant does not survive. It is not claimed here; the statements below that involve C use it only where it is present.

The logical reading is classical. Assign to each constant its principal type: $I : A \rightarrow A$, $B : (B \rightarrow C) \rightarrow (A \rightarrow B) \rightarrow A \rightarrow C$, $C : (A \rightarrow B \rightarrow C) \rightarrow B \rightarrow A \rightarrow C$, $K : A \rightarrow B \rightarrow A$, $W : (A \rightarrow A \rightarrow B) \rightarrow A \rightarrow B$. For a set $P \subseteq \{W, K, C\}$ containing C , a formula is a theorem of the implicational logic whose structural rules are exactly those of P if and only if it is the type of a closed combination of I , B , and the members of P (Curry and Feys, 1958; Hindley and Seldin, 2008). The four commutative cases are $\{C\}$ (linear, the closed linear terms), $\{C, K\}$ (affine), $\{C, W\}$ (relevant, the closed λ I-terms) and $\{C, K, W\}$ (unrestricted). Without C the corresponding logics are noncommutative and require a directed implication in the manner of Lambek (Lambek, 1958); those cases are discussed below as part of the permission lattice but no logical correspondence is claimed for them. Linear logic proper (Girard, 1987) adds exponential modalities, and Section 8 returns to the resemblance between those modalities and permission-bearing scopes.

3 Scoped Reduction Systems

Definition 3.1 (Permission structure). A *permission structure* consists of a finite set Λ of scope labels, a distinguished root label $\varepsilon \notin \Lambda$, and a map $\pi : \Lambda \cup \{\varepsilon\} \rightarrow \mathcal{P}(\{W, K, C\})$.

Definition 3.2 (Scoped terms). Fix a countable set of atoms. Scoped terms are generated by $t ::= a \mid c \mid \text{sc}_\ell \mid (tt) \mid \langle t \rangle_\ell$ with $c \in \{I, B, C, K, W\}$ and $\ell \in \Lambda$. A term without the constants sc_ℓ is a *core* term.

For an occurrence p in t , the *chain* of p is the sequence of labels of the brackets that contain p properly, read from the outside. The *effective permission* of p is $\pi_t^*(p) = \pi(\varepsilon) \cap \pi(\ell_1) \cap \dots \cap \pi(\ell_n)$ where $\ell_1, \dots, \ell_n$ is the chain. A scope can therefore restrict what its ancestors allow but never extend it.

A *redex* is an occurrence of a constant c of arity n together with n arguments attached by a spine of n consecutive application nodes with no bracket on the spine. This is the only place where brackets are opaque: a bracket between a head and its arguments prevents the redex from forming. Core steps are the contextual closure of the rules of Section 2 (without S), applied at redexes, and a step with head $c \in \{W, K, C\}$ at occurrence p is *admissible* when $c \in \pi_t^*(p)$; steps with head I or B are always admissible. The admissibility test is a side condition on a rewrite step, as in conditional rewriting logic (Meseguer, 1992), with the difference that the condition depends on the position of the redex and not on the instantiation of its variables. Contextual closure passes through brackets, so reduction inside a scope is ordinary reduction under the scope's effective permission. The *boundary rules* are

$$(\text{Abs}) \quad \langle u \rangle_\ell v \rightarrow \langle uv \rangle_\ell, \quad (\text{Sc}) \quad \text{sc}_\ell u \rightarrow \langle u \rangle_\ell,$$

both always admissible: Abs lets an argument enter a scope from the outside, and Sc lets a term be placed in a scope by an application. The core relation is written $\rightarrow_\pi$ and the relation with boundary rules $\rightarrow_\pi^+$.

Proposition 3.3 (Availability versus admissibility; status E). *A rule may be available in the system and fail to be admissible at a redex. Concretely, if $K \in \pi(\varepsilon)$ and $K \notin \pi(\ell)$, then $\langle Kbc \rangle_\ell$ contains an occurrence of the available combinator K at the head of a redex whose step is not admissible, and the term is normal.*

Lemma 3.4 (Monotonicity in permissions; status E). *If $\pi \leq \pi'$ pointwise, then $\rightarrow_\pi \subseteq \rightarrow_{\pi'}$ and $\rightarrow_\pi^+ \subseteq \rightarrow_{\pi'}^+$. In particular every term that is strongly normalizing under π' is strongly normalizing under π .*

Proof. Effective permissions are monotone in π , and the admissibility condition is a membership test in the effective permission. $\square$

Lemma 3.5 (Chain extension; status E). *If a chain γ' extends a chain γ , then $\pi^*(\gamma') \subseteq \pi^*(\gamma)$. Consequently every term that is R -normal at γ is R -normal at γ' , and the permitted redexes of a fixed term at γ' are among those permitted at γ .*

Proof. The effective permission of the extended chain is the effective permission of γ intersected with the permissions of the added labels. A redex is permitted when its head is non-structural or lies in the effective permission of its chain, and the set of permitted redexes of a fixed term at a chain is therefore antitone in the effective permission. $\square$

Moving a term into a bracket thus always moves it in the favorable direction for normality: a term without permitted redexes keeps that property in every enclosing scope. The lemma gives no control in the other direction, and it does not say that a term remains insensitive or that a pending absorption inside it remains eligible; those are properties of the term and of the absorption rule, and they are stated separately where they are used.

The first lemma shows that restricting permissions can only remove reductions, which yields more normal forms and fewer divergences. What it does not give is any control over confluence, the subject of the next section.

4 Confluence and the Boundary Rules

Theorem 4.1 (Confluence of the core system; status C). *For every permission structure, the core relation $\rightarrow_\pi$ on core terms is confluent.*

Proof. Label each constant occurrence with its effective permission. Formally, introduce symbols c^Q for $c \in \{I, B, C, K, W\}$ and $Q \subseteq \{W, K, C\}$ together with the application symbol, the unary bracket symbols, and the atoms, and call a labelled term *well labelled* when every c^Q occurs at a position whose effective permission is Q . Well-labelled terms correspond bijectively to core terms. Consider the term rewriting system with rules $I^Q x \rightarrow x$, $B^Q f g x \rightarrow f(gx)$ for all Q , and the rule for c^Q with $c \in \{W, K, C\}$ only when $c \in Q$. Each left-hand side is an application spine headed by one constant with distinct variables as arguments, so the system is left-linear. It has no critical pair: the root symbols of two distinct left-hand sides are different constants at the end of their spines, and a proper non-variable subterm of a left-hand

side is a shorter spine with the same head, which is an instance of no left-hand side because each symbol c^Q has a single rule of a single arity. Orthogonal systems are confluent (Terese, 2003, Ch. 4); see also Rosen (1973) and Klop (1992). It remains to see that well-labelled terms are closed under reduction. A step at a redex rearranges only the argument subterms of the redex, and they hang from the spine, which contains no bracket; every argument therefore lies in the same chain as the head both before and after the step, whether it is retained, copied, or discarded, and the constants inside the arguments keep the chains they had. Hence labels remain correct, and confluence of the labelled system gives confluence of the core system. $\square$

The proof shows what makes the core system tame: a step never changes the chain of any surviving constant. The boundary rules do change it, and the effect is decisive.

Theorem 4.2 (Boundary rules and uniqueness of normal forms; status C). *The system with the rule (Abs) is confluent if and only if $\pi(\varepsilon) \subseteq \pi(\ell)$ for every $\ell \in \Lambda$, that is, if and only if no scope restricts the root permission. The same holds for the system with (Sc). When the condition fails, uniqueness of normal forms fails: some term has two distinct normal forms.*

Proof. Suppose $\pi(\varepsilon) \subseteq \pi(\ell)$ for all ℓ . Then every effective permission equals $\pi(\varepsilon)$, so the system is the term rewriting system over the signature with a binary application symbol, the atoms and constants, a unary symbol $\langle \cdot \rangle_\ell$ for each label and a constant sc_ℓ , with the core rules for the combinators in $\pi(\varepsilon)$ together with (Abs $_\ell$) and (Sc $_\ell$) for every ℓ . Every left-hand side is left-linear. To exclude critical pairs, list the left-hand sides: $c x_1 \cdots x_n$ for each rule, $\langle u \rangle_\ell v$ for Abs, and $\text{sc}_\ell u$ for Sc. A critical pair requires unifying a non-variable subterm of one left-hand side with a left-hand side, or the whole of one with the whole of another. The non-variable subterms are the left-hand sides themselves, the shorter spines $c x_1 \cdots x_k$ with $0 \leq k < n$, the bracket term $\langle u \rangle_\ell$ inside an Abs left-hand side, and the constant sc_ℓ inside an Sc left-hand side. Every left-hand side has an application symbol at its root, and the function child of that application is, respectively, a spine ending in a constant c , a bracket symbol, or sc_ℓ . Two such terms unify only if their function children unify, and a spine ending in c unifies with neither a bracket term nor sc_ℓ , nor with a spine ending in a different constant, nor with a spine of the same constant and a different length. A bracket term or a constant sc_ℓ or c has a symbol at its root that is not application, so it unifies with no left-hand side. What remains is the trivial overlap of a rule with itself at the root. Hence there is no critical pair, the system is orthogonal, and it is confluent (Terese, 2003, Ch. 4). The interactions that do occur between the combinatory rules and the boundary rules take place at variable positions, since the arguments x_i are variables and no left-hand side lies inside them: for instance an Abs redex $\langle u \rangle_\ell v$ may sit inside an argument of a K redex, or a combinatory redex may sit inside u , and such overlaps are harmless in an orthogonal system. The term $\langle \text{K a} \rangle_\ell \text{b}$ shows the sequential interaction: it contains no combinatory redex, the Abs step creates one, and nothing overlaps.

Conversely suppose some ℓ and some $R \in \pi(\varepsilon) \setminus \pi(\ell)$. Choose a redex u built from the constant R and distinct atoms (atoms are not constants, so they form no redex with anything), and let u' be its contractum; for example $u = \text{K b c}$ with $u' = \text{b}$, or $u = \text{W g y}$ with $u' = \text{g y y}$, or $u = \text{C g y z}$ with $u' = \text{g z y}$. Consider $t = \langle \text{a} \rangle_\ell u$. By Abs, $t \rightarrow \langle \text{a} u \rangle_\ell$, and this term is normal: the redex u now lies in the chain ℓ where R is not admissible, and the atom a creates no redex. On the other hand $u \rightarrow u'$ is admissible at the root, so $t \rightarrow \langle \text{a} \rangle_\ell u' \rightarrow \langle \text{a} u' \rangle_\ell$, which is also normal and differs from the first. A confluent system has at most one normal form for

each term, so the system is not confluent. For (Sc) use $\text{sc}_\ell u$ in the same way: $\text{sc}_\ell u \rightarrow \langle u \rangle_\ell$ is normal, while reducing u first gives $\langle u' \rangle_\ell$. $\square$

The theorem is a statement about confluence and uniqueness of normal forms, and not about determinism. The systems considered here are nondeterministic in the ordinary sense even when they are confluent, since a term usually contains several redexes, and confluence asserts only that the choices can be reconciled. What fails when a scope restricts the root permission is joinability: the same term reaches two different normal forms according to whether an argument is reduced before or after it is absorbed. The dependence arises because the permission governing a redex is a function of position, and a boundary rule can move a redex from a position where it is permitted to one where it is not. A theory that keeps both restrictive scopes and boundary crossing therefore has to compare sets of outcomes or has to restrict the boundary rules, and Subsection 4.1 examines the second course. Observe that Theorem 4.2 has no analogue for the core system, because of Theorem 4.1; the whole difficulty lies in the rules that transport terms across boundaries.

4.1 Restricted absorption

The converse half of Theorem 4.2 suggests permitting absorption only for arguments that cannot misbehave. This subsection works with an abstract base relation, because nothing in the argument depends on the combinatory basis. Terms are generated by $t ::= a \mid (tt) \mid \langle t \rangle_\sigma$, where the atoms include the head constants of the base relation.

Definition 4.3 (Spine-based relation). A relation $\rightarrow_R$ on terms is *spine-based* when it is given by finitely many rules, at most one for each head atom c , of arity $n_c \geq 1$, of the following form. A redex at an occurrence q is a subterm $cx_1 \cdots x_n$ whose n application nodes carry no bracket on the function-child chain from q down to c ; it is rewritten to $\rho_c(x_1, \dots, x_n)$, where ρ_c is a term built from the variables x_i , application nodes, and atoms, with no bracket; and the step is permitted exactly when a predicate adm_c holds of the chain of q , the sequence of labels of the brackets that contain q . Steps are closed under all contexts, including brackets.

The combinatory rules of Section 2 with the effective-permission test of Section 3 are spine-based, and so is the scheme $((\text{p}x)y) \rightarrow x$ with any predicate on chains. The three properties used below are that the form of a redex and the admissibility of its step depend only on the spine and the chain of its root, that a contractum places each argument at the chain of the redex root, and that no left-hand side contains a bracket.

Fix a spine-based $\rightarrow_R$ and a chain of the site at which absorption is considered. A term v is *R-normal at a chain* when it contains no redex whose step is permitted at the chain inherited from its position, and *settled* at a chain when it is *R-normal* and, in addition, no application node inside it has a bracket as its function child. Informally, a settled term is irreducible and has no pending absorptions. Two restrictions of absorption are defined by a side condition on the argument:

$$(\text{Abs}_{\text{NF}}) \quad \langle u \rangle_\sigma v \rightarrow_A \langle uv \rangle_\sigma \text{ if } v \text{ is } R\text{-normal}, \quad (\text{Abs}_S) \quad \langle u \rangle_\sigma v \rightarrow_{A_S} \langle uv \rangle_\sigma \text{ if } v \text{ is settled},$$

the chain in each case being that of the application node being rewritten. Normality is thus global in the sense that it concerns every position inside v with the permissions inherited from the actual position. Settledness is exactly normality for the combined relation: a term that is

R -normal and contains an application with a bracket as function child contains an innermost such application, whose argument is then R -normal and has no such application inside it, hence is settled, so the application is an (Abs_S) redex.

Theorem 4.4 (Absorbing R -normal arguments is insufficient; status E). *Let $\rightarrow_R$ be the spine-based relation with the single rule $((\mathbf{p} x) y) \rightarrow x$, permitted at a chain γ exactly when the label τ does not occur in γ . Then $\rightarrow_R$ is terminating and confluent, but $\rightarrow_R \cup \rightarrow_A$ with absorption restricted to R -normal arguments has a term with two distinct normal forms.*

Proof. The relation $\rightarrow_R$ decreases size, and its only left-hand side has no overlap with itself, so it is terminating and confluent. Take distinct atoms $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and labels σ, τ , and let

$$t = \langle \mathbf{c} \rangle_\tau (\langle \mathbf{p} \mathbf{a} \rangle_\sigma \mathbf{b}).$$

The argument $v = \langle \mathbf{p} \mathbf{a} \rangle_\sigma \mathbf{b}$ contains no redex, because the application of $\mathbf{p}$ to $\mathbf{a}$ has one argument and the application of the bracket to $\mathbf{b}$ is not a spine. So absorption of v into τ is permitted and gives $\langle \mathbf{c} (\langle \mathbf{p} \mathbf{a} \rangle_\sigma \mathbf{b}) \rangle_\tau$. Inside this term the inner application of $\langle \mathbf{p} \mathbf{a} \rangle_\sigma$ to the normal atom $\mathbf{b}$ is absorbed, giving $N_1 = \langle \mathbf{c} \langle \mathbf{p} \mathbf{a} \mathbf{b} \rangle_\sigma \rangle_\tau$. The only redex of N_1 is $\mathbf{p} \mathbf{a} \mathbf{b}$ at the chain (τ, σ) , where the step is not permitted, and N_1 contains no absorption redex, so N_1 is normal. The other route absorbs $\mathbf{b}$ into σ first, which is permitted because $\mathbf{b}$ is normal, giving $\langle \mathbf{c} \rangle_\tau \langle \mathbf{p} \mathbf{a} \mathbf{b} \rangle_\sigma$. Now the argument has the redex $\mathbf{p} \mathbf{a} \mathbf{b}$ at the chain (σ) , where the step is permitted, so absorption is blocked until it fires: $\langle \mathbf{c} \rangle_\tau \langle \mathbf{p} \mathbf{a} \mathbf{b} \rangle_\sigma \rightarrow_R \langle \mathbf{c} \rangle_\tau \langle \mathbf{a} \rangle_\sigma \rightarrow_A \langle \mathbf{c} \langle \mathbf{a} \rangle_\sigma \rangle_\tau = N_2$, which is normal. The term t contains no redex other than the two absorptions available at the root and inside the argument, the labels σ and τ are distinct, and the only redexes in the reducts are the ones described, so every reduction sequence from t ends in N_1 or N_2 . Since $N_1 \neq N_2$, the term t has two normal forms. $\square$

The failure is an overlap of two absorptions, namely the absorption of v into τ and a pending absorption inside v , and not a conflict between absorption and the base relation. Absorbing v transports it into a chain with fewer permissions before the absorption pending inside it has exposed its redex, so that redex is evaluated under the wrong permissions. This is the obstruction that the following theorem removes.

Lemma 4.5 (Absorption of settled arguments is confluent; status E). *The relation $\rightarrow_{A_S}$ is terminating and confluent.*

Proof. For a bracket node B let $f(B)$ be the number of application nodes above B whose function-child subtree contains B , and let $\mu(t)$ be the sum of $f(B)$ over the bracket nodes of t . An absorption step replaces $\langle u \rangle_\sigma v$ by $\langle u v \rangle_\sigma$. Every node outside the redex keeps its ancestors above the redex, and the position of the redex does not change. The bracket node itself loses exactly one such ancestor, the application $\langle u \rangle_\sigma v$ that it has moved above. Brackets inside u lose the top application and gain the new application $u v$, in which u is the function child, so their count is unchanged. Brackets inside v have an application ancestor with v as the argument child both before and after, which is not counted. Brackets elsewhere are unaffected. So μ decreases by exactly one, and the relation terminates. For local confluence consider two distinct redexes. If they are disjoint the steps commute. If one is inside the function part u of the other, the inner step does not change the argument v , so the outer redex survives with the same side condition, and the two orders give the same term. Neither redex can lie inside the argument v of the other, because a settled argument has no application with a bracket as function child, which every redex has. By Newman's lemma the relation is confluent. $\square$

Lemma 4.6 (Strong commutation; status E). *Let $\rightarrow_R$ be spine-based. If $x \rightarrow_R y$ and $x \rightarrow_{A_S} z$, then there is a term w with $y \rightarrow_{A_S}^* w$ and $z \rightarrow_{\bar{R}} w$, that is, z reaches w in at most one R -step.*

Proof. Let the absorption redex be $p = \langle u \rangle_\sigma v$ with v settled at its chain, and let the R -redex be at the occurrence q with spine $cx_1 \cdots x_n$. The node p is not a spine node, because the function-child chain of a spine ends in a constant and not in a bracket, and $q \neq p$ for the same reason. There are four cases. If q is disjoint from p , the steps commute. If q lies inside v , this is impossible, because v is R -normal at its chain. If q lies inside u , then y is obtained by replacing u by u' ; the argument v is unchanged, so $y \rightarrow_{A_S} \langle u' v \rangle_\sigma$, and the same R -step is available in $z = \langle u v \rangle_\sigma$, because the redex $cx_1 \cdots x_n$ is untouched and its chain is the same, giving the same term. If q lies above p , the node p lies in an argument x_i , and, writing x'_i for the result of absorbing in x_i , the absorption step inside x_i does not touch the spine of the redex, whose nodes lie above x_i , and does not change the chain of q , so z contains the redex $cx_1 \cdots x'_i \cdots x_n$ at the same occurrence with the same spine and chain, and since the form and permission of an R -step depend only on these, the step is available and gives $w = C[\rho_c(x_1, \dots, x'_i, \dots, x_n)]$. In $y = C[\rho_c(x_1, \dots, x_n)]$ the argument x_i occurs $k \geq 0$ times, in pairwise disjoint positions, each at the chain of q because ρ_c contains no bracket. In x the argument x_i also sat at the chain of q , because the spine contains no bracket, so each copy of p has exactly the chain it had in x . Each copy is therefore an absorption redex with the same settled argument, and k absorption steps turn y into w . If $k = 0$ then $w = y$. $\square$

Theorem 4.7 (Settled absorption preserves confluence; status C). *Let $\rightarrow_R$ be spine-based and confluent. Then $\rightarrow_R \cup \rightarrow_{A_S}$ is confluent. In particular, for the combinatory systems of Section 3, settled absorption yields a confluent system for every permission structure, with no condition on π .*

Proof. By hypothesis $\rightarrow_R$ is confluent and by Lemma 4.5 so is $\rightarrow_{A_S}$. Lemma 4.6 says that the two relations strongly commute in the sense of Huet (Huet, 1980), which implies that they commute. The Hindley–Rosen lemma (Terese, 2003, Ch. 1) then gives confluence of the union. For the combinatory systems the base relation is confluent by Theorem 4.1. $\square$

Remark 4.8. The normal forms of $\rightarrow_R \cup \rightarrow_{A_S}$ are exactly the R -normal terms in which no application has a bracket as its function child, by the observation that settledness is normality for the combined relation. Settled absorption is conservative in a visible way: an argument with a pending absorption, or an argument that is still reducible, is never absorbed until it has been resolved, and an argument whose reduction diverges is never absorbed. The criterion of the Hindley–Rosen argument is recorded in the form needed: strong commutation requires that the single-step side be the R -side, because an R -step that duplicates an absorption redex is matched by several absorption steps on the other side.

Insensitive absorption

Settledness excludes two different things: permitted redexes in the argument, and pending absorptions. The counterexample of Theorem 4.4 needs both a pending absorption and a redex whose permission depends on the chain, which is created by that absorption. This suggests keeping pending absorptions and excluding permission-sensitive redexes only. In the combinatory systems of Section 3 call I and B *insensitive*, since a step headed by either is permitted at every chain, and call W, K, and C *sensitive*. Write $\rightarrow_R = \rightarrow_{R_n} \cup \rightarrow_{R_s}$ for the

corresponding split of the base relation, and call a term *insensitive* if it contains no sensitive constant. Define

$$(\text{Abs}_U) \quad \langle u \rangle_\sigma v \rightarrow_{A_U} \langle u v \rangle_\sigma \text{ if } v \text{ is } R\text{-normal and either settled or insensitive.}$$

Then $A_S \subsetneq A_U \subsetneq A_{NF}$ as sets of permitted transfers. The first inclusion is strict because $v = \langle B a \rangle_\ell b$ is R -normal and insensitive and has a pending absorption, so it is accepted by A_U and not by A_S . The second is strict because $v = \langle K a \rangle_\ell b$ is R -normal and has both a sensitive constant and a pending absorption, so A_{NF} accepts it and A_U does not; this is the argument of the counterexample of Theorem 4.4 with K in the role of p . Settled arguments with sensitive constants, such as $K a$, are accepted by A_U as before.

Theorem 4.9 (Insensitive absorption preserves confluence; status C). *For the scoped combinatory systems of Section 3, with an arbitrary permission structure, $\rightarrow_R \cup \rightarrow_{A_U}$ is confluent.*

Proof. Let $X = \rightarrow_{R_n} \cup \rightarrow_{A_U}$, so that $\rightarrow_R \cup \rightarrow_{A_U} = \rightarrow_{R_s} \cup X$. The plan is to show that X is confluent, that $\rightarrow_{R_s}$ is confluent, and that $\rightarrow_{R_s}$ strongly commutes with X .

X terminates. Order terms lexicographically by size and then by the quantity μ of Lemma 4.5. A step headed by I or B strictly decreases size, and an absorption step keeps the size and decreases μ by one.

X is locally confluent. Consider two distinct redexes of X in a term. Two R_n redexes give a peak of the orthogonal system $\{I, B\}$, which closes by R_n steps. Consider an R_n redex and an absorption redex $p = \langle u \rangle_\sigma v$ with v R -normal. An R_n step cannot lie inside v , which is R -normal. If it lies inside u , or disjoint from p , the two steps commute, because the side condition concerns v only. If it lies above p , then p lies in an argument of the redex, which I and B keep exactly once and at the chain of the redex root, so the steps commute with one step on each side. Consider two absorption redexes $p_1 = \langle u_1 \rangle_{\sigma_1} v_1$ and p_2 . If they are disjoint, or p_2 lies in u_1 , the steps commute. The remaining case is p_2 inside v_1 , symmetric cases being exchanged. Then v_1 has an application with a bracket as function child, so it is not settled; hence it is insensitive, and so is everything inside it. Let x contain p_1 , let y be obtained by absorbing at p_2 inside v_1 , giving v'_1 , and let z be obtained by absorbing at p_1 . The term v'_1 is insensitive. Let v^* be an X -normal form of v'_1 , which exists by termination. It is R -normal, since it has no insensitive redex and no sensitive constant, and it is insensitive, so absorption into σ_1 is permitted. Hence $y \rightarrow_X^* \langle u_1 \rangle_{\sigma_1} v^* \rightarrow_{A_U} \langle u_1 v^* \rangle_{\sigma_1}$. In $z = \langle u_1 v_1 \rangle_{\sigma_1}$ the redex p_2 is still present, now at a longer chain, and absorbing there is permitted, because the side condition on its argument is R -normality of an insensitive term, which does not depend on the chain. The sequence of steps from v'_1 to v^* can be replayed at the longer chain by the following invariant. Every X -reduct of an insensitive term is insensitive, because I and B steps and absorptions rearrange existing subterms and create no constant. For an insensitive term w there is no sensitive constant, so the R -redexes of w are exactly the steps headed by I or B , which are permitted at every chain, and R -normality of w is the absence of such redexes. In particular, for an insensitive argument w' occurring in w , the side condition of A_U (that w' is R -normal, and settled or insensitive, with insensitivity holding) is the same at every chain. Hence the set of X -steps available in an insensitive term, together with their side conditions, is independent of the chain at which the term sits. This is a property of the combinatory systems, and it does not follow from Lemma 3.5, which provides only that R -normality survives extension of the chain; for a spine-based relation with chain-dependent admissibility the replay would need an

additional hypothesis. So the steps from v'_1 to v^* are available at the chain extended by σ_1 , and $z \rightarrow_X^* \langle u_1 v^* \rangle_{\sigma_1}$ as well. By Newman's lemma X is confluent.

$\rightarrow_{R_s}$ is confluent, as a subsystem of the orthogonal system of Theorem 4.1.

Strong commutation. Let $x \rightarrow_{R_s} y$ and $x \rightarrow_X z$. If the second step is an R_n step, this is a peak of two steps of an orthogonal system: the redexes are disjoint, or one lies in an argument of the other. An R_s step duplicating the R_n redex is matched by several R_n steps on the side of y and one R_s step on the side of z ; an R_s step erasing it gives $z \rightarrow_{R_s} y$; and an R_n step with an R_s redex in an argument gives one step on each side. If the second step is an absorption at $p = \langle u \rangle_\sigma v$, then the R_s redex does not lie inside v , which is R -normal, and the cases of inside u , disjoint, and above p are those of Lemma 4.6: in the last case each of the k copies of p in y has the chain of p in x , because the contractum has no bracket and x_i sat at the chain of the redex root, the side condition on v depends only on v and that chain, and $z \rightarrow_{R_s}^{\bar{}} w$ with $y \rightarrow_{A_U}^k w$. Hence $y \rightarrow_X^* w$ and $z \rightarrow_{R_s}^{\bar{}} w$.

By the Hindley–Rosen lemma (Terese, 2003, Ch. 1) and strong commutation (Huet, 1980), $\rightarrow_{R_s} \cup X$ is confluent. $\square$

The two properties that the discussion of transport suggests keeping apart are visible in the proof. Internal stability is the statement that the steps inside the transported argument do not depend on the chain; for an insensitive argument this holds because its steps are permitted at every chain, and for a settled argument because it has none. Interface stability is the statement that reductions created by composing the argument with the receiving term remain joinable with those available before the transfer; the redexes created at the interface lie on the spine of uv and outside v , and they are evaluated at the same chain whichever of the two competing steps is taken first, which is what the case analysis above verifies. The theorem does not cover arguments with sensitive constants and pending absorptions, since for those internal stability can fail, as the counterexample of Theorem 4.4 shows.

Conjecture 4.10 (Transport-invariant absorption; status P). *Call v transport-invariant between two chains if every term v' reachable from v by reductions internal to v , that is, by steps of $\rightarrow_R \cup \rightarrow_A$ taking place inside v without involving any enclosing context or receiving term, has the same set of permitted R -redexes at the two chains. Then absorption restricted to transport-invariant arguments, in place of settled or insensitive ones, also preserves confluence for confluent spine-based relations.*

Proposition 4.11 (A decidable criterion for safe transfer; status E). *For the scoped combinatory systems of Section 3, which have a fixed finite permission alphabet, define $\text{Safe}(v, \gamma)$ to hold when v is R -normal at the chain γ and is either settled or insensitive. Given the effective permission of γ , which is computed by $|\gamma|$ intersections of bitmasks, $\text{Safe}(v, \gamma)$ is decidable in time $O(|v|)$. It does not depend on the receiving function u or on the label σ beyond the chain. If every absorption of a system satisfies Safe , that is, if the system is $\rightarrow_R \cup \rightarrow_{A_U}$, then the system is confluent by Theorem 4.9. The criterion is conservative: it rejects every argument that still contains a permitted redex, and every argument that has both a sensitive constant and a pending absorption, including arguments that Conjecture 4.10 would admit.*

Proof. A single traversal of v detects permitted redexes by spine shape and one membership test in the effective permission of the chain, detects application nodes with a bracket as function child, and detects sensitive constants. The confluence claim restates Theorem 4.9. The claim concerns the system whose absorptions are exactly the safe ones: confluence of a system

in which some absorptions fire without the check is not asserted, and neither is joinability of a single accepted transfer inside such a system. For a spine-based relation outside the combinatory systems the same traversal bound holds with each constant-time test replaced by the cost of one admissibility test, and only the settled disjunct is covered by Theorem 4.7. $\square$

Three properties of increasing strength are therefore separated. Normality of the transferred object is a local syntactic condition, and Theorem 4.4 shows that it does not suffice. Preservation of admissibility under transfer concerns the effect of the boundary on permitted transformations, and it is the content of Conjecture 4.10; in the permission-intersection calculus settledness implies it, because a settled term has no permitted redex at the source chain and therefore none at a more restricted chain, while redexes created by attaching the term to a function lie outside it. Joinability of competing executions is the global semantic property, and Theorems 4.7 and 4.9 derive it from the first condition strengthened either by the absence of pending absorptions or by the absence of permission-sensitive constants. A deterministic evaluation strategy would select one execution without restoring confluence, so when a transfer fails the criterion the alternatives are to delay it, to prohibit it, to retain several outcomes, or to record enough history to tell them apart. Applications to code motion, interface boundaries, and permission systems are suggested by this separation and are not developed here.

In the permission-intersection calculus of Section 3, where moving a term inward can only remove permissions (Lemma 3.5), every settled argument is transport-invariant: it has no permitted redex at the source chain, hence none at the more restricted target chain, and none can appear inside it, since the only new redexes created by attaching it to u lie on the spine of uv and not inside v . For an arbitrary spine-based relation whose admissibility predicates are not monotone in the chain this inference needs the extra assumption that the argument is also settled at the target chain. The converse fails, since arguments with permitted redexes can be transport-invariant. The counterexample of Theorem 4.4 is consistent with the conjecture, since its argument is not transport-invariant: the redex $\mathbf{pab}$ reachable from v is permitted at the root chain and not at (τ) . The conjecture would admit absorption of arguments that still contain redexes, provided moving them cannot change whether those redexes are permitted. It is stated as a proposal because the proof of Lemma 4.6 used in an essential way that an absorbable argument has no steps of its own.

5 Copy and Reference in Scoped Term Graphs

The distinction between copying an expression and copying a reference to a shared expression is the second family of questions, and it is the structural counterpart of the contrast between call-by-name duplication and call-by-need sharing (Wadsworth, 1971; Ariola and Felleisen, 1997). Represent a scoped term as a finite rooted directed acyclic graph with ordered successors. A node carries an atom, a constant, an application label with a function successor and an argument successor, a bracket label ℓ with one successor, or an indirection with one successor, and a node may have several parents. Indirections are followed whenever a successor is inspected; write $\text{res}(m)$ for the node reached from m by following indirections. The *unraveling* $U(m)$ is the scoped tree defined by $U(m) = m$ at an atom or constant, $U(m) = U(f)U(x)$ at an application node with successors f and x , $U(m) = \langle U(s) \rangle_\ell$ at a bracket node with successor s , and $U(m) = U(s)$ at an indirection with successor s ; write $U(G) = U(\text{root})$. It is finite because G is finite and acyclic, and $U(m)$ depends only on the part of G below m . Term graph

rewriting in this sense is treated by Barendregt et al. (1987).

A *redex node* is an application node $n = n_k$ whose spine $n_k, n_{k-1}, \dots, n_0$, defined by taking n_{j-1} to be the node reached from the function successor of n_j by res, consists of application nodes $n_k, \dots, n_1$ and a constant node n_0 of arity k , and contains no bracket node. Write x_j for the argument successor of n_j . The intermediate spine nodes $n_1, \dots, n_{k-1}$ may be shared with other parents. A *core graph step* at n overwrites the node n alone and leaves every other node unchanged, so the other parents of the intermediate spine nodes continue to see the partial applications they saw before. For I and K the node n becomes an indirection to x_1 . For B it becomes an application with function successor x_1 and argument a fresh application node with successors x_2 and x_3 . For C it becomes an application whose function successor is a fresh application node with successors x_1 and x_3 and whose argument is x_2 . For W it becomes an application whose function successor is a fresh application node with successors x_1 and x_2 and whose argument is x_2 , so that x_2 acquires a second incoming edge. No argument node is copied, every parent of n sees the contractum, and a shared redex is contracted once. Nodes that become unreachable are discarded. The step preserves acyclicity, because the successors of the new n and of the fresh nodes are among $x_1, \dots, x_k$ and the fresh nodes, all of which lie strictly below n or are fresh. The step for (Sc) at an application node n whose function successor resolves to the constant sc_ℓ and whose argument successor is u overwrites n by a bracket node with label ℓ and successor u ; it is used in Example 5.4 and not in the theorems below. Absorption steps on graphs are defined in Section 5.1. A reduction sequence is any finite sequence of steps, so the sets of normal forms defined below are existential: they collect the normal forms reached by some sequence.

An *occurrence* of a node n in $U(m)$ is a path from m to n through successor edges, equivalently the position in $U(m)$ that the path determines. Two occurrences of n are disjoint, since an occurrence of n inside another would be a path from n to n , which a finite acyclic graph does not have.

Lemma 5.1 (Unraveling of core steps; status E). *Let G' be obtained from the acyclic graph G by a core step at the redex node n . For every node m of G that survives in G' , the tree $U_{G'}(m)$ is obtained from $U_G(m)$ by contracting, simultaneously, every occurrence of n in $U_G(m)$ as a redex of the tree system. Each occurrence is a redex occurrence of the tree system whose chain is the chain of the corresponding path.*

Proof. Induct on the height of m in G . If $m = n$, then $U_G(n) = cU(x_1) \cdots U(x_k)$, with the intermediate spine nodes contributing only their own unravelings, and it has the single occurrence of n at its root and none inside it. The unraveling $U_{G'}(n)$ is $\rho_c(U(x_1), \dots, U(x_k))$ by the definition of the contractum, since the fresh nodes have successors among the x_j and the nodes x_j are unchanged. If m does not reach n , nothing below m changed and there is no occurrence. If $m \neq n$ reaches n , then $U(m)$ is built from the unravelings of the successors of m by the same constructor in G and in G' , because only n and fresh nodes differ, and the occurrences of n in $U_G(m)$ are exactly the occurrences in the unravelings of the successors. The induction hypothesis applies to each successor. The occurrence is a redex of the tree system because its spine has no bracket, by definition of a redex node, and its chain is the list of brackets along the path. $\square$

The permission governing a redex now depends on the path by which it is reached. Define the *context set* $\text{Ctx}(n)$ of a node as the set of effective permissions $\pi(\varepsilon) \cap \pi(\ell_1) \cap \cdots \cap \pi(\ell_k)$

over all root paths to n , where $\ell_1, \dots, \ell_k$ are the labels of the bracket nodes passed. A node is *coherent* if $\text{Ctx}(n)$ has one element, and a graph is coherent if every node is. Two policies for admitting a step at a redex node n with head $R \in \{W, K, C\}$ are natural: policy ANY admits it if R lies in some member of $\text{Ctx}(n)$, and policy ALL admits it if R lies in every member. They agree on coherent nodes. For a graph G and a policy write $\text{NF}_{\text{pol}}(G)$ for the set of unravelings of normal graphs reachable from G , and write $\text{NF}_{\text{tree}}(t)$ for the set of normal forms reachable from the tree t in the scoped tree system. The observation model throughout is this set of normal forms, with scope labels visible; sharing itself is not observed, because U discards it. A model in which the shape of the sharing is observable would distinguish copies from references even when no scope boundary is crossed, and is a different theorem.

Theorem 5.2 (Adequacy under coherence; status C). *Let G be a coherent core graph. (a) Every core graph step from G yields a coherent graph G' , and $\text{Ctx}_{G'}(m) \subseteq \text{Ctx}_G(m)$ for every surviving node m . (b) The policies ANY and ALL permit exactly the same steps at G and at every graph reachable from G . (c) If $G \rightarrow^* N$ with N normal, then $U(N)$ is the unique normal form of $U(G)$; hence $\text{NF}_{\text{pol}}(G) \subseteq \text{NF}_{\text{tree}}(U(G))$, a set with at most one element.*

Proof. (a) Let the step be at the redex node n , with $\text{Ctx}(n) = \{Q\}$. A root path in G' to a surviving node m either avoids the nodes created or overwritten by the step, in which case it is a path of G , or it passes through n . In the second case it continues from n through application nodes of the contractum, which are not brackets, to an argument node x_i , and then through x_i to m . The path of G that reaches n , runs down the spine of the redex, which contains no bracket, to the same x_i , and continues identically crosses the same brackets, so it has the same context. Hence $\text{Ctx}_{G'}(m) \subseteq \text{Ctx}_G(m)$, a singleton, and the set is nonempty because m is still reachable. The extra incoming edge that W gives to x_2 is of exactly this kind: it comes from a node below n . Fresh nodes are reachable only through n and see only Q .

(b) On a coherent graph each $\text{Ctx}(n)$ is a singleton, so membership in some member and in every member coincide, and by (a) coherence persists along any sequence of steps.

(c) By Lemma 5.1 the tree $U(G')$ is obtained from $U(G)$ by contracting all occurrences of n , which are pairwise disjoint redex occurrences of the tree system. By coherence they have the same effective permission, the unique member of $\text{Ctx}(n)$, so each is permitted exactly when the graph step is. Contracting them one after another is a reduction sequence of the tree system, because contracting an occurrence leaves the chain and shape of each disjoint occurrence unchanged. Thus $U(G) \rightarrow^+ U(G')$ for every step. A normal graph has no permitted redex node, and by Lemma 5.1 every redex occurrence of $U(N)$ lies over a redex node, so $U(N)$ has no permitted redex and is normal. Uniqueness follows from Theorem 4.1. $\square$

The agreement of the full sets of normal forms needs a statement about normalizing strategies, which is recorded separately.

Proposition 5.3 (Adequacy of graph normalization; status P). *Let G be a coherent core graph such that $U(G)$ has a normal form. Then $\text{NF}_{\text{pol}}(G) = \text{NF}_{\text{tree}}(U(G))$.*

This is a proposal. The intended derivation uses the adequacy theorem for orthogonal term rewriting systems of Kennaway et al. (1994) in the following form: for an orthogonal system and an acyclic term graph, graph reduction by in-place contraction with sharing is adequate for term reduction, in that if the unraveling has a normal form then a normalizing graph strategy reaches a graph whose unraveling is that normal form. The statement of that theorem has to

be matched against the present setting before the proposition can be promoted: the system must be presented as an orthogonal labelled system, in which each constant carries its effective permission, and this is well defined on nodes by coherence, with the labels remaining correct along reduction by part (a) of Theorem 5.2. The graphs are acyclic and the contraction convention is in-place with sharing preserved. Until that matching is carried out, only the inclusion of part (c) is claimed.

Coherence is therefore sufficient for copy and reference to agree in the sense of part (c), under the observation model above. It is not a property of the syntax of the starting term. A tree is trivially coherent, and the graph of a tree-shaped term can acquire incoherent sharing along its own reduction.

Example 5.4 (Reachable incoherence). Let $\Lambda = \{\ell_1, \ell_2\}$ with $\pi(\varepsilon) = \{W, K, C\}$, $\pi(\ell_1) = \{K\}$, $\pi(\ell_2) = \emptyset$, and use the core rules together with (Sc). Put $h = B(CB\text{sc}_{\ell_2})(Ba\text{sc}_{\ell_1})$. Routine reduction gives $huv \rightarrow a(\text{sc}_{\ell_1}u)(\text{sc}_{\ell_2}v)$: indeed $hu \rightarrow CB\text{sc}_{\ell_2}(Ba\text{sc}_{\ell_1}u)$, which reduces to $B(a(\text{sc}_{\ell_1}u)\text{sc}_{\ell_2})$, and applying this to v gives the claim. Let $t = Whx$ with $x = Kbc$, a tree. Define $N_1 = a\langle b \rangle_{\ell_1} \langle b \rangle_{\ell_2}$, $N_2 = a\langle b \rangle_{\ell_1} \langle Kbc \rangle_{\ell_2}$, and $N_3 = a\langle Kbc \rangle_{\ell_1} \langle Kbc \rangle_{\ell_2}$.

Proposition 5.5 (status E). *In Example 5.4, $\text{NF}_{\text{tree}}(t) = \{N_1, N_2\}$, $\text{NF}_{\text{Any}}(G_t) = \{N_1\}$, and $\text{NF}_{\text{All}}(G_t) = \{N_1, N_3\}$.*

Proof. In the tree system the W step produces hxx , which reduces by the displayed computation to $a(\text{sc}_{\ell_1}x)(\text{sc}_{\ell_2}x)$, in which each copy sits at the root until its sc_{ℓ_i} is applied. While a copy is at the root, K is admissible and it may reduce to b ; once wrapped, the copy in ℓ_1 can still reduce because $K \in \pi(\ell_1)$, so it always ends as b , while the copy in ℓ_2 ends as b if it was reduced before being wrapped and stays Kbc otherwise. That gives N_1 and N_2 . The term N_3 is not a tree normal form, because the copy inside ℓ_1 keeps a permitted K -redex ($K \in \pi(\ell_1)$) and the system continues to N_1 or N_2 ; it is a normal form only of the graph under ALL, where the shared node is blocked by its other context. In the graph the W step shares x and the two wrapping steps give x two parents, one inside each scope, so after both wrappings $\text{Ctx}(x) = \{\{K\}, \emptyset\}$. Under ANY the redex at x remains admissible in every configuration, because the root context and the context of ℓ_1 both contain K , so x is always eventually reduced and the only normal form is N_1 . Under ALL the redex is admissible until a wrapping in ℓ_2 has occurred and blocked afterwards. Reducing x before that wrapping gives N_1 ; wrapping in ℓ_2 first and leaving x unreduced gives a normal graph whose unraveling is N_3 . The form N_2 would require the copy in ℓ_1 to reduce while the copy in ℓ_2 does not, which is impossible when the two are one node. $\square$

The proposition shows that once incoherence appears the three semantics separate in two distinct ways. Sharing suppresses normal forms that independent copying makes available: both graph policies miss N_2 , because the two references to the shared node cannot be reduced independently, and this incompleteness is the effect of reduction across distinct contexts. In addition, ALL is unsound relative to the tree semantics in the sense of normal forms. The term N_3 is reachable in the tree system, since both copies may be wrapped before either is reduced, but it is not a normal form there: the copy inside ℓ_1 keeps a permitted K -redex and the system continues. In the graph the single node is also referenced from ℓ_2 , whose permission lacks K , so under ALL the node is blocked and the graph is stuck with N_3 as a normal form that no tree reduction ends in. The shared node has no single permission to act under, and every choice

of convention changes the answer. The multiplicity of tree normal forms here is an instance of Theorem 4.2, since the example uses (Sc) with restrictive scopes.

Incoherence does not always change the answer, which is why the condition is not necessary.

Proposition 5.6 (status E). *Replace $x = \mathbf{Kbc}$ in Example 5.4 by $x = \mathbf{lb}$. The reachable graph is again incoherent, but $\text{NF}_{\text{tree}} = \text{NF}_{\text{Any}} = \text{NF}_{\text{All}} = \{N_1\}$.*

Proof. The only redex inside the shared node is a step with head $\mathbf{l}$, which requires no permission, so every policy admits it in every context. $\square$

The shared computation is pure, terminating, and insensitive to sharing, as the proposition requires, and cross-boundary duplication is there nonetheless. A sharper sufficient condition than coherence would ask only that every redex reachable inside a shared node be permission-neutral or have the same admissibility in all contexts, but its formulation needs the reachability notion of the next section.

5.1 Absorption in graphs

Theorem 5.2 concerns core steps, which leave every chain unchanged. Absorption changes chains, so its behavior on shared nodes needs its own analysis. The question is whether a transfer authorized along one access path may be performed on a node that other paths also reach. Write $\rightarrow_U$ for the tree relation $\rightarrow_R \cup \rightarrow_{A_U}$ of Theorem 4.9, and for an effective permission q write $\text{Safe}(v, q)$ when the tree v is R -normal at q and either settled or insensitive. In this subsection graphs are finite and acyclic, and the policy for core steps is ALL: a core step at n with head $R \in \{\mathbf{W}, \mathbf{K}, \mathbf{C}\}$ is admitted when R lies in every member of $\text{Ctx}(n)$, and a step headed by $\mathbf{l}$ or $\mathbf{B}$ is always admitted.

Let n be an application node whose function successor resolves to a bracket node b with label ℓ and successor u , and whose argument successor resolves to v . The *in-place absorption* at n overwrites n by a bracket node with label ℓ whose successor is a fresh application node with successors u and v ; the node b is not modified, so other parents of b are unaffected. A *root path* to n is a sequence $\varpi = ((m_0, i_0), \dots, (m_{r-1}, i_{r-1}), m_r)$ with m_0 the root, $m_r = n$, each m_j for $j < r$ an application or bracket node, and m_{j+1} the resolved i_j -th successor of m_j . It determines a position $\text{pos}(\varpi)$ in $U(G)$, and its effective permission $q(\varpi)$ is $\pi(\varepsilon)$ intersected with $\pi(\ell')$ over the labels ℓ' of the bracket nodes among $m_0, \dots, m_{r-1}$; the set $\text{Ctx}(n)$ is the set of these over all root paths. The *path-local absorption* $\text{copyAbs}(G, \varpi)$ adds $r + 2$ fresh nodes: copies $m'_0, \dots, m'_{r-1}$, a bracket node b' with label ℓ , and an application node a' with successors u and v that is the successor of b' . The copy m'_j has the label of m_j and the successors of m_j , except that the i_j -th successor is m'_{j+1} when $j + 1 < r$ and b' when $j + 1 = r$. The new root is m'_0 , or b' if $r = 0$. Every other node, including n and b , is unchanged and may become unreachable, and every successor of a copy that is off the path continues to point at the original shared node. Both operations preserve acyclicity, since the new edges lead to fresh nodes or to nodes already below n . In-place absorption is a rewriting of the node n , and every parent of n sees the result; path-local absorption rewrites a single incoming path and leaves the other references of n as they were.

Lemma 5.7 (Unraveling of absorption steps; status E). *(a) If G' is obtained from G by in-place absorption at n , then for every surviving node m the tree $U_{G'}(m)$ is obtained from $U_G(m)$ by replacing every occurrence of n , simultaneously, by $\langle U(u) U(v) \rangle_\ell$; the occurrences are pairwise*

disjoint, and each is an absorption redex $\langle U(u) \rangle_\ell U(v)$ of the tree system whose chain is the chain of its path. (b) For a root path ϖ to n ,

$$U(\text{copyAbs}(G, \varpi)) = U(G)[\text{pos}(\varpi) \leftarrow \langle U(u) U(v) \rangle_\ell].$$

Proof. (a) The argument of Lemma 5.1 applies, with $U_{G'}(n) = \langle U(u) U(v) \rangle_\ell$ and $U_G(n) = \langle U(u) \rangle_\ell U(v)$. (b) Only the root changes, and the nodes of the new graph that are reachable from the new root are the copies, b' , a' , and original nodes. Induct downward along the path: $U(b') = \langle U(u) U(v) \rangle_\ell$, and for $j < r$ the node m'_j has the label and successors of m_j except the one on the path, which leads to a node whose unraveling is the unraveling of m_{j+1} with the subtree at the remaining path replaced. Hence $U(m'_j)$ is $U(m_j)$ with the subtree at the remaining path replaced, and for $j = 0$ this is the claim. $\square$

Theorem 5.8 (Graph-transfer soundness; status E). *Let G be a finite acyclic scoped graph. (a) If $G \rightarrow G'$ is a core step admitted under ALL, then $U(G) \rightarrow_R^+ U(G')$. (b) If $G' = \text{copyAbs}(G, \varpi)$ and $\text{Safe}(U(v), q(\varpi))$ holds, then $U(G) \rightarrow_{AU} U(G')$, the step being at $\text{pos}(\varpi)$. (c) If G' is obtained by in-place absorption at n and $\text{Safe}(U(v), q)$ holds for every $q \in \text{Ctx}(n)$, then $U(G) \rightarrow_{AU}^+ U(G')$.*

Proof. (a) By Lemma 5.1 the tree $U(G')$ arises from $U(G)$ by contracting the disjoint occurrences of n , each with chain whose effective permission is a member of $\text{Ctx}(n)$. Under ALL the head is permitted at every member, so each contraction is a permitted tree step. Contracting an occurrence does not change the chain or the shape of a disjoint occurrence, so the contractions can be performed one after another, and the contractum is built from the argument nodes, which is why the result is $U(G')$. (b) By Lemma 5.7(b) the tree $U(G')$ differs from $U(G)$ exactly by replacing $\langle U(u) \rangle_\ell U(v)$ at $\text{pos}(\varpi)$ by $\langle U(u) U(v) \rangle_\ell$, which is an absorption step; its side condition is $\text{Safe}(U(v), q(\varpi))$, since the chain at $\text{pos}(\varpi)$ is the list of brackets along ϖ . (c) By Lemma 5.7(a) the tree $U(G')$ arises by replacing the disjoint absorption redexes at the occurrences of n . The chain of each occurrence has its effective permission in $\text{Ctx}(n)$, so the side condition holds at each, and a step at one occurrence changes neither the chain nor the argument of a disjoint one. $\square$

Corollary 5.9 (status C). *Let $G \rightarrow^* N$ be a finite sequence of steps, each of which is a core step admitted under ALL, a path-local absorption satisfying the side condition of Theorem 5.8(b), or an in-place absorption satisfying the side condition of Theorem 5.8(c). If $U(N)$ is normal for $\rightarrow_U$, then $U(N)$ is the unique $\rightarrow_U$ -normal form of $U(G)$.*

Proof. Theorem 5.8 gives $U(G) \rightarrow_U^* U(N)$, and $\rightarrow_U$ is confluent by Theorem 4.9, so a normal form reachable from $U(G)$ is unique. $\square$

The side conditions of Theorem 5.8 are sufficient, and the following examples show that they cannot be dropped from a general statement. They do not show that every sound step satisfies them: a particular unsafe step can happen to produce a reachable tree.

Proposition 5.10 (Failures without the sufficient conditions; status E). *(a) Core steps under ANY are not sound: in the wrapped graph of Example 5.4, whose unraveling is N_3 , a core step under ANY at the shared node yields a graph unraveling to N_1 , which is not reachable from N_3 . (b) In-place absorption is not sound when safety holds at one member of $\text{Ctx}(n)$ and fails at another: there are a graph G and a node n for which the result of in-place absorption*

is not reachable from $U(G)$ by $\rightarrow_U$. (c) If $U(v)$ is insensitive, then $\text{Safe}(U(v), q)$ does not depend on q , so the condition of Theorem 5.8(c) is checked at a single chain and no coherence hypothesis is needed. (d) Absorption can create incoherence: there is a coherent graph that has an incoherent node after an absorption, in place or path-local.

Proof. (a) From N_3 the only permitted tree step is the K-step inside ℓ_1 , since $\pi(\ell_2) = \emptyset$, and it leads to N_2 , a tree normal form. The shared node has contexts $\{K\}$ and $\emptyset$, and under ANY the step at it is admitted and gives both references the contractum b , so the graph unravels to N_1 , which is not reachable from N_3 .

(b) Let $\Lambda = \{\sigma, \ell\}$ with $\pi(\varepsilon) = \pi(\sigma) = \{W, K, C\}$ and $\pi(\ell) = \emptyset$. Let $v = K b c$, let n be the node $\langle c \rangle_\sigma v$, and let G be the graph of $d \ n \ \langle n \rangle_\ell$ in which n is shared. Then $\text{Ctx}(n) = \{\{W, K, C\}, \emptyset\}$. At the empty permission v has no permitted redex and no pending absorption, so Safe holds; at the full permission v is a permitted redex, so Safe fails. In-place absorption gives a graph unraveling to $T = d \ \langle c v \rangle_\sigma \ \langle \langle c v \rangle_\sigma \rangle_\ell$. The term $U(G) = d \ (\langle c \rangle_\sigma v) \ \langle \langle c \rangle_\sigma v \rangle_\ell$ has exactly two kinds of step: the absorption at the occurrence inside ℓ , which is permitted, and the K-step at the occurrence of v at the root position, because K is not permitted inside ℓ . The absorption at the root-position occurrence is not permitted while that v is unreduced, since Safe fails there. The target T contains the unreduced v inside σ at the root position, so reaching it would require absorbing there before the K-step; and any sequence that performs the K-step at that occurrence replaces v by b , which no step restores. So T is not reachable. By contrast, the path-local absorption along the path through ℓ satisfies Safe at the empty permission and yields $d \ (\langle c \rangle_\sigma v) \ \langle \langle c v \rangle_\sigma \rangle_\ell$, a single $\rightarrow_U$ step.

(c) For an insensitive term the permitted R -redexes are the steps headed by I or B at every chain, and the settled-or-insensitive disjunct holds without reference to the chain.

(d) Let $\pi(\sigma) \subsetneq \pi(\varepsilon)$ and let G be the graph of $d \ (\langle c \rangle_\sigma e) e$ with the atom e shared. Both references to e lie at the root chain, so G is coherent. After absorption of e into σ , in place or path-local, the node e is below the bracket σ along one path and at the root along the other, so $\text{Ctx}(e) = \{\pi(\varepsilon) \cap \pi(\sigma), \pi(\varepsilon)\}$ has two members. $\square$

The proposition separates sharing computation from sharing authority, and the difference is that between a union and an intersection over the contexts of a node: $R \in \bigcup \text{Ctx}(n)$ says that the step is permitted along some access path, $R \in \bigcap \text{Ctx}(n)$ says that it is permitted along every one, and the first does not imply the second. Policy ANY tests the union and policy ALL the intersection, and part (a) is the failure of the first as a test for an operation that every reference will see. Core reduction, and the absorption of insensitive arguments, can be performed once on a shared node, because the result does not depend on which reference reaches it. Absorption of an argument whose admissible reductions depend on the chain changes the context of computation. In-place absorption is sound on a shared node when the transfer is authorized along every reference; path-local absorption is one way to perform a transfer that is authorized along a single reference without altering the others, and it is a sufficient implementation and not a required one. The permission test for the in-place form is weaker than coherence: it asks for safety at every member of $\text{Ctx}(n)$ and not for equality of all effective permissions. Only soundness is claimed. The converse, that every tree reduction is simulated by a graph reduction using path-local copying, and the agreement of normal forms, are not treated here.

6 Descent of Admissibility Under Reduction

The classification of Section 2 attaches a structural fragment to a system. The scoped setting suggests attaching one to a term, and the question is whether the scope tree of the term determines it. Say that an occurrence q' in a reduct is a *descendant* of an occurrence q of a constant in the original term if it is a copy of it obtained along the reduction; the rules never create constants other than by copying, and the constants sc_ℓ disappear when used, so every constant occurrence in a reduct descends from one in the original term.

Lemma 6.1 (Descent; status E). *If $t \rightarrow_\pi^+ t'$ and q' is a descendant of q , then $\pi_{t'}^*(q') \subseteq \pi_t^*(q)$.*

Proof. By the argument of Theorem 4.1, a core step leaves the chain of every surviving or copied constant unchanged. A step by Abs on $\langle u \rangle_\ell v$ leaves the chains inside u unchanged and appends ℓ to the chain of every occurrence inside v . A step by Sc on $\text{sc}_\ell u$ appends ℓ to the chain of every occurrence inside u . In every case the chain is extended or unchanged, and the effective permission is an intersection over the chain, so it can only shrink. $\square$

Define $\text{Adm}(t)$ as the set of $R \in \{W, K, C\}$ for which t contains an occurrence of the constant R whose effective permission contains R , and define the *realized fragment* $\text{Real}(t)$ as the set of R for which some reduction sequence from t contains an admissible step with head R . The first is computed from the scope tree and the positions of the constants. The second is computed over the reachable configurations.

Theorem 6.2 (Static bound; status E). *$\text{Real}(t) \subseteq \text{Adm}(t)$ for every scoped term t .*

Proof. An admissible step with head R fires at an occurrence q' of R in some reduct t' , with $R \in \pi_{t'}^*(q')$. By the remark above q' descends from an occurrence q of R in t , and by Lemma 6.1 applied along the sequence, $\pi_t^*(q) \supseteq \pi_{t'}^*(q') \ni R$. Hence $R \in \text{Adm}(t)$. $\square$

The bound answers the question of whether the scope tree alone classifies a term, and the answer is that it classifies it only from above.

Proposition 6.3 (status E). *(a) The bound is not tight: for $K \in \pi(\varepsilon) \cap \pi(\ell)$, the term $\langle K \rangle_\ell$ has $\text{Adm} = \{K\}$ and $\text{Real} = \emptyset$. (b) The fragment admissible at the redexes currently present, $\sigma(t)$, is neither monotone increasing nor monotone decreasing along reduction: with $K \in \pi(\varepsilon) \cap \pi(\ell)$, the term $\langle K a \rangle_\ell b$ has $\sigma = \emptyset$, its Abs-reduct $\langle K a b \rangle_\ell$ has $\sigma = \{K\}$, and the next step yields $\langle a \rangle_\ell$ with $\sigma = \emptyset$. (c) The admissibility of a given occurrence can strictly drop: with $K \in \pi(\varepsilon) \setminus \pi(\ell)$, the occurrence of K in $\langle a \rangle_\ell (K b c)$ has effective permission containing K , while its descendant in the Abs-reduct $\langle a (K b c) \rangle_\ell$ does not.*

Proof. Each claim is a direct computation. In (a) the term is a single bracketed constant with no arguments, so no step is possible. In (b) a spine of one argument is not a K -redex, the reduct has a full spine inside a scope that permits K , and contracting it leaves no constants. In (c) the chain of the occurrence changes from empty to (ℓ) and $K \notin \pi(\ell)$. $\square$

The consequence is that a classification of a term as ordered, linear, affine, relevant, or unrestricted cannot be read from its scope tree. What can be said is that the initial occurrence-wise permissions bound the fragment the term can ever use, that the fragment realized along a particular reduction can be strictly smaller, and that the descent in Lemma 6.1 is the only movement: no rule extrudes a term from a scope, so permission is never regained. A dynamic

classification, over reachable configurations, is thus the right notion, and the realized fragment is the invariant it computes. Whether $\text{Real}(t)$ is decidable, even for the core system, is not settled here.

7 Boundary-Governed Transition Systems

The preceding sections are an instance of a more general setting in which the possibility of a transition, its admissibility, its observable effect, and its recorded standing are separate structures. A *boundary-governed system* is a tuple $\mathfrak{X} = (P, E, s, t, \mathcal{A})$ in which P is a set of configurations, E a set of elementary transitions with source and target maps $s, t : E \rightarrow P$, and $\mathcal{A}(x) \subseteq s^{-1}(x)$ the transitions admissible at x . Admissibility is a function of the configuration and not of the transition alone, which is the form of dependence that Section 3 requires: the same redex is admissible in one configuration and inadmissible in another. A history is a finite composable sequence of transitions, that is, a morphism of the free path category on (P, E) (Mac Lane, 1998, Ch. II), and an observation is a functor out of the category generated by admissible transitions. Two distinct admissible histories with the same endpoints and the same observation show that the observation is not faithful; this holds immediately from the definition of faithfulness (Mac Lane, 1998, Ch. I), and it remains a statement about functors, since an event that changes a record without changing the visible state establishes non-faithfulness only when it has been specified how events become morphisms and how observation acts on them. A finer notion of history that separates causal dependence from mere sequence is supplied by event structures (Winskel, 1987), and the free path category is the coarsest history semantics in that sense.

For the core scoped system take P to be the set of core terms, E the set of pairs of a term and a redex occurrence in it, whether or not the step is permitted, and $\mathcal{A}(t)$ the redex occurrences whose head is permitted at the effective permission of the occurrence. The target $\mathfrak{X}'$ is the unscoped system on bracket-free terms with permission $\pi(\varepsilon)$, and q erases brackets, sending a redex occurrence of t to the corresponding occurrence of $q(t)$. An abstraction $q = (q_P, q_E)$ between boundary-governed systems commutes with source and target. Three properties of the admissible sets are distinct. Preservation asks $q_E(\mathcal{A}(x)) \subseteq \mathcal{A}'(q_P(x))$. Reflection asks that a concrete transition at x whose image is abstractly admissible be concretely admissible, that is $q_E^{-1}(\mathcal{A}'(q_P(x))) \cap s^{-1}(x) \subseteq \mathcal{A}(x)$. Lifting asks $\mathcal{A}'(q_P(x)) \subseteq q_E(\mathcal{A}(x))$, that every abstractly admissible transition has an admissible concrete representative at x . Reflection does not imply lifting, as the term $\langle l \rangle_\ell a$ of Proposition 7.3 shows. In general lifting does not imply reflection either, since several concrete transitions may share an image; for the erasure functor below, q_E is injective on redex occurrences, because a redex occurrence is determined by its root application node and q induces a bijection between the application nodes of t and those of $q(t)$, and for a transition of $q(t)$ with an exposed preimage lifting at the term does imply reflection there. Preservation is the step-level form of soundness of an abstraction, in the sense that the abstract semantics over-approximates the concrete one (Cousot and Cousot, 1977), and preservation together with lifting is the one-step shape of a simulation and, with the converse, of a bisimulation (Sangiorgi, 2011, Ch. 1).

Say that a redex occurrence r of $q(t)$ is *exposed* in t if it is the image under q_E of a redex occurrence of t ; equivalently, the application nodes of the spine of r correspond to application nodes of t joined by function-child edges with no bracket between them. A term is *transparent* if no bracket node is the function child of an application node. Erasing a boundary can create

a redex that the boundary interrupted, and this is a different phenomenon from the loss of a permission.

Proposition 7.1 (Preservation; status E). *If $t \rightarrow_\pi t'$ is a permitted core step, then $q(t) \rightarrow q(t')$ is a permitted step of $\mathfrak{X}'$ at the image redex. Hence q preserves admissibility at every core term and for every permission structure, and it maps permitted paths to permitted paths.*

Proof. A redex of t has a spine without brackets, so its erasure is the corresponding spine, and the rules contain no brackets, so q commutes with contraction:

$$q(\rho_c(x_1, \dots, x_n)) = \rho_c(q(x_1), \dots, q(x_n)).$$

Context closure is respected because q is a homomorphism on application. The effective permission of the occurrence is contained in $\pi(\varepsilon)$, so a step permitted in t is permitted in $q(t)$. $\square$

Proposition 7.2 (Reflection; status E). *Reflection holds at every core term if and only if $\pi(\varepsilon) \subseteq \pi(\ell)$ for every $\ell \in \Lambda$.*

Proof. Under the condition every effective permission equals $\pi(\varepsilon)$, so a concrete redex is permitted exactly when its image is. If $R \in \pi(\varepsilon) \setminus \pi(\ell)$, the redex of $\langle R \cdots \rangle_\ell$ built from the constant R and atoms, where as in Theorem 4.2 atoms are not constants, is a concrete transition whose image is permitted and which is not permitted. $\square$

Proposition 7.3 (Lifting and exposure; status E). *(a) An abstractly permitted transition of $q(t)$ at a redex r lifts to a permitted concrete transition at t if and only if r is exposed in t and its head is non-structural or lies in the effective permission of the corresponding occurrence. (b) If t is transparent, every redex of $q(t)$ is exposed, and if moreover $\pi(\varepsilon) \subseteq \pi(\ell)$ for all ℓ , lifting holds at t . (c) Lifting fails at $\langle l \rangle_\ell \mathbf{a}$ for every permission structure, and at $\langle K \mathbf{a} \rangle_\ell \mathbf{b}$ when $K \in \pi(\varepsilon)$, even if every scope permits every structural rule. Consequently lifting holds at every core term if and only if $\Lambda = \emptyset$: the first failure alone gives the direction that a nonempty Λ breaks lifting, and the second shows that the failure persists when every scope is fully permissive.*

Proof. (a) The preimage of r under q_E is empty unless r is exposed, in which case it is a single redex occurrence, and the permission requirement is the definition of $\mathcal{A}$. (b) Without function-position brackets no bracket lies on any spine path; the second claim follows from (a) and the equality of the effective permissions with $\pi(\varepsilon)$. (c) In $\langle l \rangle_\ell \mathbf{a}$ and $\langle K \mathbf{a} \rangle_\ell \mathbf{b}$ the bracket is the function child of an application, so the spine of the abstract redex $\mathbf{la}$, respectively $\mathbf{Ka b}$, passes through it, no concrete redex exists, and the abstract steps are permitted. $\square$

Proposition 7.4 (Lifting up to absorption; status E). *Suppose $\pi(\varepsilon) \subseteq \pi(\ell)$ for every $\ell \in \Lambda$. If $q(t) \rightarrow u$ is a permitted abstract step, then in the system with (Abs) there are a term t'' and a term t' with $t \rightarrow_{\text{Abs}}^* t'' \rightarrow_\pi t'$, $q(t'') = q(t)$ and $q(t') = u$.*

Proof. Let ρ be the root of the redex in $q(t)$ and follow the function children from the corresponding node of t down to the head. If a bracket lies on this path, take the one nearest the root; every node between it and the root is an application, and it is the function child of some application a on the path. Applying (Abs) at a moves the bracket above the new application, one level closer to ρ , or off the path if a is the root. For a fixed bracket the number of application nodes above it on the path decreases with each such step, so after finitely many steps it leaves the path; the number of brackets on the path is finite and does not increase, so the

double induction, first on the brackets on the path and then on the application nodes above the nearest one, is well founded and terminates with no bracket on the path. The terms before and after satisfy $q(t'') = q(t)$ because (Abs) does not change the erasure. Now the redex is exposed, its chain has been extended by labels, and the effective permission equals $\pi(\varepsilon)$ by hypothesis, so the abstract permission suffices, and the concrete step $t'' \rightarrow t'$ has $q(t') = u$ by Proposition 7.1. $\square$

Proposition 7.1 shows that preservation is unconditional, and Propositions 7.2 and 7.3 show that reflection and lifting fail for independent reasons. Reflection fails because a scope can withdraw a permission that the root grants. Lifting fails because a scope can interrupt a spine, even when it withdraws nothing. Exposure is moreover a dynamic property. Transparency is not preserved by reduction: the step $\mathbf{B} f \langle g \rangle_\ell x \rightarrow f \langle g \rangle_\ell x$ creates a bracket in function position, so a transparent term can reach a term whose redexes are not exposed, and the interplay with the realized fragment of Section 6 is a topic for further work. A third dimension is historical. The steps (Abs) are sent to identities by q , so distinct concrete histories can share an image: for $t = \mathbf{p} \langle g \rangle_\ell \mathbf{a} \langle h \rangle_\ell \mathbf{b}$ the two orders of the independent absorptions give distinct paths with the same endpoints and the same image, and a faithful reading of the abstraction on histories must record them as different.

The descent of Lemma 6.1 is a statement about the direction of change, and a boundary-governed system makes visible that the direction of change does not settle which changes are legitimate. Let $\kappa : P \rightarrow L$ assign to each configuration its available capabilities in a partially ordered set L , and call a transition $e : x \rightarrow y$ *attenuating* if $\kappa(y) \preceq \kappa(x)$. Let $x_0 \in P$ be an initial configuration, let $F \subseteq P$ be an endpoint predicate, and write $x_0 \rightarrow_{\mathcal{A}}^* y$ when there is a history from x_0 to y each of whose transitions is admissible at the configuration where it is taken.

Proposition 7.5 (Attenuation and endpoints do not determine authorization; status E). *There is a finite boundary-governed system with a capability map κ , an initial configuration x_0 , and an endpoint predicate F such that (a) every transition is attenuating, yet some transitions are not admissible; (b) $F(y)$ holds for a configuration y for which there is no history $x_0 \rightarrow_{\mathcal{A}}^* y$; and (c) a configuration y satisfying F is reached by one admissible history and by one history containing a non-admissible transition, so that the endpoint does not determine whether the history was authorized.*

Proof. Let $P = \{p, b, c, d\}$ and $E = \{e_1 : p \rightarrow b, e_2 : b \rightarrow c, e_3 : p \rightarrow c, e_4 : p \rightarrow d\}$, with $\kappa(p) = \{r, w\}$, $\kappa(b) = \{r\}$, and $\kappa(c) = \kappa(d) = \emptyset$ in the order of inclusion. Let $\mathcal{A}(p) = \{e_1\}$, $\mathcal{A}(b) = \{e_2\}$, and $\mathcal{A}(c) = \mathcal{A}(d) = \emptyset$, let $x_0 = p$, and let $F(y)$ hold when $\kappa(y) = \emptyset$. Each transition satisfies $\kappa(t(e)) \subseteq \kappa(s(e))$, which gives (a), and $e_3, e_4 \notin \mathcal{A}$. For (b) take $y = d$: the only transition into d is e_4 , which is not admissible. For (c) take $y = c$: the history $e_1 e_2$ is admissible and e_3 is not, and both end at c . $\square$

The proposition is the observation that a permission-lowering move is not thereby an authorized move. Lemma 6.1 establishes that every step of the scoped calculus lowers or preserves effective permission occurrence by occurrence, and that fact does not by itself say which of the lowering moves a protocol allows; in the scoped calculus the protocol is the pair of rules (Abs) and (Sc) together with the side conditions on them, and the results of Section 4 identify conditions under which distinct authorized reduction paths remain joinable. They do not make the execution deterministic, and they do not guarantee termination. It supplies a

further distinction beside availability, admissibility, exposure, and realization: a final configuration may satisfy an endpoint predicate without any authorized path from the specified initial configuration,

$$F(y) \not\Rightarrow \exists h : x_0 \rightarrow_{\mathcal{A}}^* y.$$

The same remark applies to records of histories. A validator that examines only relations internal to a supplied history, such as the consecutive matching of source and target, accepts every history with the same internal relations, so its acceptance cannot tell which of two internally consistent histories was originally committed. Binding a history to an independently retained commitment, by requiring in addition that a fixed function of the history agree with a stored value, adds a separate condition; it does not establish that the events occurred, and its strength is that of the function used.

The two boundary rules behave differently under erasure. Since $q(\langle u \rangle_\ell) = q(u)$, (Abs) is sent to an identity transition. Extend q to terms containing scope constants by the convention $q(\text{sc}_\ell) = \text{!}$; this is a translation and not bracket erasure, it enlarges the domain of the abstraction beyond the core terms, and it is introduced here as an explicit extension. Then $q(\text{sc}_\ell u) = \text{!} q(u) \rightarrow q(u) = q(\langle u \rangle_\ell)$, and an (Sc) step is sent to an actual step of ! and not to an identity; the distinction matters for the path category, in which a reduction and an identity morphism are different. Lifting for (Sc) is a separate question that is not treated here. The uniqueness of normal forms also does not pass along q in either direction. By Theorem 4.2, a scoped system with (Abs) and a restriction at the root can have terms with two normal forms although the unscoped system obtained by erasure is confluent, so q does not reflect confluence; by Theorem 4.7 this failure is removed when absorption is restricted to settled arguments.

The following is a proposal and not a result. The local-to-global question suggested by Section 5 is whether locally permitted reductions in different scopes can be glued to a global reduction. Example 5.4 shows that they cannot in general, since a node with incompatible local permissions has no global one. A sheaf-theoretic account (Johnstone, 2002, Part C) would have to construct the site and restriction maps before using this as an obstruction, and this essay does not attempt that construction.

8 Concatenative Languages, Exponentials, and Open Problems

In a stack language the words that duplicate, drop, and exchange the top items are contraction, weakening, and exchange made primitive, and quotations are program fragments passed as values. A scoped permission structure would assign such words different powers inside different quotations. The results of Sections 4 and 5 suggest what to expect: if a quotation may be absorbed into a context with fewer permissions after it has begun to evaluate, evaluation order becomes observable, and uniqueness of normal forms is lost exactly when the context restricts anything, unless absorption waits for settled arguments (Theorem 4.7).

A scope whose permission set contains both W and K behaves like a region of unrestricted resource use, and one with empty permission like a region of ordered use. This resembles the role of exponential modalities in linear logic (Girard, 1987), but the resemblance is not a theorem. A modality $!_\sigma A$ indexed by scope should be derived from a sound interpretation of the operational calculus, and the descent of Lemma 6.1 indicates what such an interpretation must respect, since typing judgments for a term inside a scope cannot rely on permissions that a later absorption would remove.

The problems left open are the following. The independence of C from $\{I, B, K, W\}$ is not established above and would complete the separation of the three structural axes. Whether the realized fragment $\text{Real}(t)$ is decidable, for the core system or for the system with boundary rules, is unknown. Absorption restricted to arguments normal for the base relation is shown not to restore confluence (Theorem 4.4), absorption restricted to settled arguments, or to insensitive ones, is shown to do so (Theorems 4.7 and 4.9), and whether the settledness condition can be weakened to transport-invariance (Conjecture 4.10) is unsettled, as is the exact class of boundary rules that preserve coherence of term graphs along reduction. A necessary and sufficient condition for the agreement of copy and reference semantics, in place of the sufficient coherence condition of Theorem 5.2, would require an observation model fine enough to detect resource consumption and reduction traces, and the three sets in Proposition 5.5 suggest a family of such models ordered by what they can distinguish. Distinct histories with the same endpoints suggest 2-cells between them, and the coherence questions that follow belong to the theory of bicategories (Lack, 2010). The reachable terms that are transparent, and the interaction of exposure with the realized fragment, are not characterized. The normalization-adequacy statement of Proposition 5.3 awaits a check of the hypotheses of the cited theorem. For graph absorption, completeness of path-local copying with respect to the tree relation and the agreement of normal forms are open, and so is a notion of coherence relative to the constants a node actually contains. Finally, the abstraction of Section 7 should be extended to histories and observation functors, so that the preservation, reflection, and lifting properties are stated not only for single steps but for admissible paths.

References

- Z. M. Ariola and M. Felleisen. The call-by-need lambda calculus. *Journal of Functional Programming* 7(3):265–301, 1997.
- H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Second edition. North-Holland, 1984.
- H. P. Barendregt, M. C. J. D. van Eekelen, J. R. W. Glauert, J. R. Kennaway, M. J. Plasmeijer, and M. R. Sleep. Term graph rewriting. In *PARLE: Parallel Architectures and Languages Europe*, Lecture Notes in Computer Science 259. Springer, 1987.
- A. Church. *The Calculi of Lambda-Conversion*. Princeton University Press, 1941.
- P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM Symposium on Principles of Programming Languages*, pages 238–252. ACM, 1977.
- H. B. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland, 1958.
- J.-Y. Girard. Linear logic. *Theoretical Computer Science* 50(1):1–102, 1987.
- J. R. Hindley and J. P. Seldin. *Lambda-Calculus and Combinators: An Introduction*. Cambridge University Press, 2008.
- G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM* 27(4):797–821, 1980.

- P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, Oxford Logic Guides 43–44. Oxford University Press, 2002.
- J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. On the adequacy of graph rewriting for simulating term rewriting. *ACM Transactions on Programming Languages and Systems* 16(3):493–523, 1994.
- J. W. Klop. Term rewriting systems. In *Handbook of Logic in Computer Science*, volume 2, pages 1–116. Oxford University Press, 1992.
- S. Lack. A 2-categories companion. In *Towards Higher Categories*, IMA Volumes in Mathematics and its Applications 152, pages 105–191. Springer, 2010.
- J. Lambek. The mathematics of sentence structure. *American Mathematical Monthly* 65(3):154–170, 1958.
- S. Mac Lane. *Categories for the Working Mathematician*. Second edition, Graduate Texts in Mathematics 5. Springer, 1998.
- J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science* 96(1):73–155, 1992.
- S. Pestov, D. Ehrenberg, and J. Groff. Factor: a dynamic stack-based programming language. In *Proceedings of the 6th Symposium on Dynamic Languages*. ACM, 2010.
- B. K. Rosen. Tree-manipulating systems and Church–Rosser theorems. *Journal of the ACM* 20(1):160–187, 1973.
- D. Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.
- M. Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen* 92:305–316, 1924.
- Terese. *Term Rewriting Systems*, Cambridge Tracts in Theoretical Computer Science 55. Cambridge University Press, 2003.
- C. P. Wadsworth. *Semantics and Pragmatics of the Lambda-Calculus*. DPhil thesis, University of Oxford, 1971.
- G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, Lecture Notes in Computer Science 255, pages 325–392. Springer, 1987.