Presentation Theory VII: Assembly Envelopes and Operational Fibres

Luca Blanchi
June 2026

ABSTRACT

This paper develops the part of Presentation Theory concerned with operations. The basic point is not that an operation has an overhead bound. Rather, a family of guarded operations generates a new presentation system: its records are construction histories, circuits, diagrams, decompositions, or verification-bearing expressions. This assembly envelope has its own image filtration, operational fibres, bridge geometry, search cost, audit cost, coherence cost, and observable lower-bound theory. The main reusable theorem is an observable barrier principle: a method based on a declared family of observables can prove that an object is outside a budgeted operational image only when those observables separate the target from that image; any residual indistinguishable low-budget object is a barrier for that method. Two applications are included to keep the formalism anchored: tensor rank and border rank as exact and closure assembly complexity, and proof-carrying split-extension records for semidirect products.

INTRODUCTION

The earlier parts of Presentation Theory study description systems, transfer, normal forms, verifiable access, fibre geometry, and observables. Operations require a separate layer only if they are treated as more than maps with overhead. A map

sigma:X_1timescdotstimes X_nto Y

can certainly give an upper bound

mathbf C_Y(sigma(x_1,ldots,x_n))
le
F(mathbf C_X_1(x_1),ldots,mathbf C_X_n(x_n)).

That statement alone is not enough for a new core module.

The stronger claim is that operations generate access. Once an operation is declared, an object of Y may be presented by a record explaining how it was assembled. Such records have hidden parameters, guards, verification data, equivalent rewritings, decomposition fibres, and inverse problems. For example, a semidirect product is not an operation of two variables:

(N,H)mapsto Nrtimes H.

The action is part of the operation:

(N,H,varphi)mapsto Nrtimes_varphi H,
qquad
varphi:HtoAut(N).

Similarly, a tensor-rank decomposition is not merely a tensor; it is a construction record expressing the tensor as a sum of rank-one pieces.

This paper isolates the formal core:

- the assembly envelope generated by guarded operations;
- operational image filtrations and assembly fibres;
- the distinction between producing, verifying, searching, and navigating records;
- observable barriers for lower-bound methods;
- two concrete applications, tensor decompositions and split-extension records.

The terminology is intentionally modest. The paper does not claim new tensor-rank lower bounds or new

group-isomorphism algorithms. It gives a precise presentation-theoretic calculus for operational constructions and for the barriers faced by lower-bound methods that only see specified observables.

GUARDED OPERATIONAL SIGNATURES

Definition (Typed presentation base).
A typed presentation base consists of a set of sorts S and, for every sin S, a presentation system

Gamma_s=(D_s,rho_s,mathbf C_s),

where D_s is a class of descriptions, rho_s:D_sto X_s is a realization map, and mathbf C_s:D_sto B_s is a cost function into an ordered budget domain.

Definition (Guarded operation).
A guarded operation of type

sigma:s_1,ldots,s_nto t

consists of:

• a semantic partial operation

sigma_X:X_s_1timescdotstimes X_s_ndashrightarrow X_t;

• a guard predicate Guard_sigma specifying when the operation is defined;
• a parameter space P_sigma, containing any action, gluing datum, quotient datum, basis choice, limit datum, or verification data required by the operation;
• a description-level compiler

widehatsigma:D_s_1timescdotstimes D_s_ntimes P_sigmato D_t;

• a nondecreasing overhead profile F_sigma.

The compiler is sound if, whenever the guard and verification data are accepted,

rho_t(widehatsigma(d_1,ldots,d_n,p))
=
sigma_X(rho_s_1(d_1),ldots,rho_s_n(d_n),rho(p)).

In pseudo, categorical, or approximate contexts the equality is replaced by the declared isomorphism or error guarantee.

Remark.
The parameter space is not optional bookkeeping. It prevents a false operation from hiding essential data. Quotients require normality data, tensor products require compatible module structures, pushouts require diagrams, and semidirect products require actions.

Definition (Operational ledger).
For an operational record r, the ledger may include several coordinates:

mathbf L(r)=
(mathbf P(r),mathbf G(r),mathbf V(r),mathbf S(r),mathbf K(r),mathbf O(r)),

standing respectively for production, guard checking, verification, search, coherence, and observation. A scalar cost is a chosen monotone function of this ledger, but the separate coordinates should not be identified unless a theorem justifies doing so.

ASSEMBLY ENVELOPES

Definition (Record format).
A record format mathcal F specifies how operation records are built. Typical formats are trees, directed acyclic graphs, circuits, diagrams, and verification-bearing records. The choice matters: sharing a subrecord may be impossible in a tree and cheap in a DAG.

Definition (Assembly envelope).
Let Gamma=\Gamma_s_sin S, let Sigma be a guarded operational signature, and let mathcal F be a record format. The assembly envelope

Ass_Sigma,mathcal F(Gamma)

is the presentation system whose descriptions are the well-formed mathcal F-records generated from base descriptions and operations in Sigma, whose realization map is obtained by evaluating the record, and whose cost is the declared ledger cost of the record.

Theorem (Existence of the assembly envelope).
Every typed presentation base, guarded operational signature, and record format with sound local compilers determines an assembly presentation system

Ass_Sigma,mathcal F(Gamma)
=
(D_ass,rho_ass,mathbf C_ass).

Proof.
The descriptions D_ass are generated inductively. Base descriptions are records. If r_i are records of sorts s_i, and if pin P_sigma supplies accepted guard and verification data for an operation sigma:s_1,ldots,s_nto t, then

sigma(r_1,ldots,r_n;p)

is a record of sort t. The realization map is defined by structural recursion: base records realize by the original rho_s, and compound records realize by applying the semantic operation to the realizations of their children and to the realized parameter. Soundness of the local compiler ensures that the compiled description realizes the same object. The cost is the ledger cost assigned recursively by the format and overhead profiles. Thus the records, realization map, and cost form a presentation system.
■

Proposition (Conservative inclusion).
Each base presentation system Gamma_s maps into the corresponding sort of the assembly envelope by viewing a base description as a length-zero assembly record. Hence assembly complexity is always relative to the chosen primitives and may be lower than native presentation complexity only because additional operational access has been declared.

Proof.
The inclusion sends din D_s to the base record d. Its realization is unchanged, and its cost is the base ledger cost. Any lower cost in the envelope must therefore come from records using operations in Sigma, not from a change in the represented object.
■

OPERATIONAL IMAGES AND ASSEMBLY FIBRES

Definition (Operational assembly cost).
For an object y in a target sort, define

mathbf C_Sigma,mathcal F(y)
=
inf\mathbf C_ass(r):rho_ass(r)=y\.

This is not an absolute complexity. It depends on the base systems, the operations, the record format, and the ledger.

Definition (Budgeted operational image).
The budgeted operational image is

Im_Sigma,mathcal F(b)
=
\y:mathbf C_Sigma,mathcal F(y)le b\.

The sets Im_Sigma,mathcal F(b) form an increasing filtration of the target sort.

Definition (Assembly fibre).
The assembly fibre of y at budget b is

mathcal A_Sigma,mathcal F(y;b)
=
\r:rho_ass(r)=y, mathbf C_ass(r)le b\.

The unrestricted fibre is the union over all budgets.

Theorem (Inverse problems are fibre problems).
The following tasks are tasks about mathcal A_Sigma,mathcal F(y;b):

• existence of a construction of y within budget b;
• search for a construction record of minimal or bounded cost;
• enumeration of all bounded construction records;
• uniqueness or non-uniqueness of decompositions;
• navigation between two construction records of the same object;
• reconstruction of a record from partial observables.

Proof.
Each task is a statement about the set of records realizing y. Existence asks whether the fibre is nonempty. Search asks for an element of the fibre. Enumeration asks for the whole bounded fibre. Uniqueness asks for its cardinality or for its quotient under a bridge relation. Navigation asks for paths inside the fibre. Reconstruction from observables asks which subsets of the fibre are consistent with the observed data.
■

VERIFICATION, SEARCH, AND COHERENCE

Definition (Verification system).
A verification system for operational records is a predicate

Check(r,pi)in\0,1\,

where pi is verification data. It is sound if Check(r,pi)=1 implies that r is a semantically valid record realizing its claimed object. It is complete on a class if every valid record in that class has accepted verification data.

Proposition (Bounded search decidability).
Assume that records of cost at most b are effectively finite, that realization equality against a target y is decidable or soundly checkable on this bounded set, and that the verification system is sound and complete on the bounded class. Then the question

mathcal A_Sigma,mathcal F(y;b)nevarnothing

is decidable by bounded search.

Proof.
Enumerate all records of cost at most b. For each record, enumerate or compute its verification data within the bounded complete format, and test whether it realizes y. Soundness prevents false positives. Completeness and effective finiteness ensure that, if a bounded valid record exists, the search finds it.
■

Definition (Bridge system and bridge cost).
A bridge system is a collection of moves between records that preserve realization. The bridge cost between two records is

mathbf B(r,r')
=
inf\cost of a bridge chain from r to r'\.

If no chain is available, mathbf B(r,r')=infty.

Proposition (Semantic equality does not imply cheap coherence).
Without an explicit bridge completeness and cost-control hypothesis, there is no function of record cost alone that bounds mathbf B(r,r') for records with rho_ass(r)=rho_ass(r').

Proof.
Take a presentation system with two records r_n,r'_n of cost 1 realizing the same object. Declare the bridge graph between them to be a path of length n, or declare no path at all. The semantic realization and record costs are fixed, but the bridge cost is n or infinite. Hence a cost bound requires bridge hypotheses; it does not follow from semantic equality.
■

OBSERVABLE BARRIERS

Definition (Observable method).
Let Omega be a family of observables on the target sort. A finite observable packet LambdasubseteqOmega defines

Phi_Lambda(y)=(omega(y))_omegainLambda.

An Omega-only lower-bound method is one whose final separation claim has the form

Phi_Lambda(y)notin Phi_Lambda(Im_Sigma,mathcal F(b))

for some finite packet Lambda of declared cost.

Theorem (Observable lower-bound soundness).
If an observable packet Lambda satisfies

Phi_Lambda(y)notin Phi_Lambda(Im_Sigma,mathcal F(b)),

then

mathbf C_Sigma,mathcal F(y)>b.

Proof.
If mathbf C_Sigma,mathcal F(y)le b, then yinIm_Sigma,mathcal F(b). Therefore Phi_Lambda(y) belongs to the image Phi_Lambda(Im_Sigma,mathcal F(b)), contradicting the separation assumption.
■

Definition (Observable residual fibre).
For a target y, budget b, and observable packet Lambda, the residual fibre is

Res_Lambda(y;b)
=
\zinIm_Sigma,mathcal F(b):
Phi_Lambda(z)=Phi_Lambda(y)\.

Theorem (Barrier as residual fibre).
If Res_Lambda(y;b)nevarnothing, then no lower-bound proof whose final test uses only the packet Lambda can establish mathbf C_Sigma,mathcal F(y)>b. Conversely, if Res_Lambda(y;b)=varnothing and the image Phi_Lambda(Im_Sigma,mathcal F(b)) is effectively certified, then the packet gives a sound Lambda-only proof of the lower bound.

Proof.
If zinRes_Lambda(y;b), then z has cost at most b and has the same packet value as y. Thus the packet cannot separate y from the budgeted image. Any proof whose final semantic claim is exactly such a packet separation therefore fails. Conversely, if the residual fibre is empty, then no budget-b object has the same packet value as y. A verification record for the packet image verifies

Phi_Lambda(y)notin Phi_Lambda(Im_Sigma,mathcal F(b)),

and the observable lower-bound theorem gives the result.
■

Remark.
The converse is deliberately effective only under an image-certification hypothesis. Set-theoretic separation is not automatically an auditable proof.

TENSOR RANK AND OBSERVABLE LOWER-BOUND METHODS

Let A,B,C be finite-dimensional vector spaces over a field k, and let

Tin Aotimes Botimes C.

Definition (Rank assembly envelope).
The rank assembly envelope has rank-one operation

(a,b,c)mapsto aotimes botimes c

and addition of tensors. A rank record of length r is a list

(a_i,b_i,c_i)_i=1^r

realizing

sum_i=1^r a_iotimes b_iotimes c_i.

The primary cost coordinate is r.

Theorem (Tensor rank as assembly complexity).
In the rank assembly envelope, the assembly cost of T in the summand coordinate is its tensor rank:

mathbf C_rank(T)=R(T).

Proof.
A record of cost r is precisely an expression of T as a sum of r rank-one tensors. Minimizing r over all such records is exactly the definition of tensor rank.
■

Definition (Closure assembly).
Over k=mathbb C, the closure rank envelope allows parameterized rank-r records whose realized tensors converge to T. The resulting cost is the least r such that T lies in the Zariski closure of the rank-le r image.

Theorem (Border rank as closure assembly complexity).
The closure assembly cost of T is its border rank:

mathbf C_border(T)=underline R(T).

Proof.
The image of rank-r assembly is the set of tensors of rank at most r. Its Zariski closure is the r-th secant variety of the Segre variety. The least r for which T belongs to that closure is, by definition, the border rank.
■

Definition (Tensor lower-bound observable).
A tensor lower-bound observable is an invariant omega such that the values of omega on rank-le r or border-rank-le r tensors satisfy a checkable constraint. Flattening ranks, Young flattenings, equations of secant varieties, and representation-theoretic tests are examples when a concrete format and checker are declared.

Corollary (Method barriers for tensor rank).
Let Lambda be a finite packet of tensor observables. If there exists a tensor Z of rank at most r such that

Phi_Lambda(Z)=Phi_Lambda(T),

then no Lambda-only method can prove R(T)>r. The same statement holds for border rank after replacing rank-le r by its closure.

Proof.
This is the residual-fibre theorem applied to the rank or border-rank assembly image.
■

Proposition (Sound tensor verification records).
In an exact computable field model, an upper-bound record for R(T)le r is verified by expanding the r rank-one summands and comparing coordinates with T. A lower-bound record using an observable packet Lambda is sound when it verifies both the value Phi_Lambda(T) and the exclusion

Phi_Lambda(T)notinPhi_Lambda(Rank_le r).

Proof.
For the upper bound, coordinate expansion computes the tensor realized by the record; equality proves the claim. For the lower bound, the observable lower-bound theorem proves soundness once the exclusion is verified. Border-rank upper bounds require an additional limit or closure verification format; the same principle applies, but the checker is algebraic rather than only coordinate expansion.
■

SPLIT EXTENSIONS AND SEMIDIRECT ASSEMBLY

Definition (Semidirect record).
A semidirect assembly record consists of groups N,H, an action

varphi:HtoAut(N),

and verification data proving that N and H are groups, that each varphi(h) is an automorphism of N, and that varphi is a homomorphism. The realized object is

Nrtimes_varphiH.

Proposition (Semidirect assembly soundness).

If a semidirect record passes the declared guard checks, then the multiplication

(n,h)(n',h')=(nvarphi(h)(n'),hh')

defines a group on Ntimes H.

Proof.
Associativity follows from associativity in N and H, from the fact that each varphi(h) is an automorphism, and from the homomorphism identity

varphi(hh')=varphi(h)varphi(h').

The identity is (e_N,e_H), and the inverse of (n,h) is

(varphi(h^-1)(n^-1),h^-1).

Thus the record realizes a group.
■

Definition (Split assembly fibre).
For a finite group G, the split assembly fibre consists of all semidirect records (N,H,varphi,theta), where

theta:Nrtimes_varphiHto G

is an isomorphism in the declared verification format. Its bounded subfibre contains only records of bounded ledger cost.

Theorem (Marked split-isomorphism criterion).
Let G=Nrtimes_varphiH and G'=N'rtimes_varphi'H'. A factor-preserving isomorphism of split records of the form

F(n,h)=(alpha(n),beta(h))

exists if and only if alpha:Nto N' and beta:Hto H' are group isomorphisms satisfying

alphacircvarphi(h)circalpha^-1
=
varphi'(beta(h))
quadfor all hin H.

Proof.
Assume F(n,h)=(alpha(n),beta(h)) is a homomorphism. Comparing

F((e,h)(n,e))
quadandquad
F(e,h)F(n,e)

gives

alpha(varphi(h)(n))=varphi'(beta(h))(alpha(n)),

which is the displayed compatibility. Conversely, if the compatibility holds, then

F((n,h)(n',h'))
=
F(nvarphi(h)(n'),hh')

equals

(alpha(n)varphi'(beta(h))(alpha(n')),beta(h)beta(h'))
=
F(n,h)F(n',h').

Thus F is a group isomorphism preserving the marked factors.
■

Corollary (Proof-carrying split-isomorphism records).
A record containing alpha,beta, verification data that they are isomorphisms, and verification data for the action-compatibility equation is a sound proof-carrying record for a marked split isomorphism.

Proof.
The checker verifies the hypotheses of the marked split-isomorphism criterion. The induced map F(n,h)=(alpha(n),beta(h)) is then an isomorphism by the theorem.
■

Remark.
The unmarked group-isomorphism problem is harder. An isomorphism may move the chosen kernel or complement. In the language of this paper, the unmarked problem is a search problem over a larger split assembly fibre, together with bridge moves changing complements and actions. The paper does not claim to solve that search problem; it isolates the verification and fibre structure of split records.

WHAT THE CALCULUS CLAIMS

The calculus developed here is intentionally limited. It does not make tensor rank easy, does not produce new matrix-multiplication algorithms, and does not solve group isomorphism. It supplies a formal layer for questions of the following kind:

- Which primitive operations are allowed?
- What records do they generate?
- Which objects lie in the budgeted operational image?
- What is the fibre of all constructions of a given object?
- What does it cost to verify a record?
- What does it cost to find a record?
- Which observables can separate a target from the budgeted image?
- Which residual fibres are barriers for a chosen method?

This is the sense in which operations are presentation engines. They do not merely combine presented objects; they create new presentation systems with their own lower bounds, inverse problems, audit formats, and residual obstructions.

REFERENCES

•
K. Efremenko, A. Garg, R. Oliveira, and A. Wigderson.
Barriers for rank methods in arithmetic complexity.
9th Innovations in Theoretical Computer Science Conference, 2018.

•
J. A. Grochow and Y. Qiao.
Algorithms for group isomorphism via group extensions and cohomology.
SIAM Journal on Computing 46 (2017), 1153--1216.

•
J. M. Landsberg.
Tensors: Geometry and Applications.
Graduate Studies in Mathematics 128, American Mathematical Society, 2012.

•
G. Ottaviani and P. Reichenbach.
Tensor rank and complexity.
arXiv:2004.01492.

•
A. Seress.
Permutation Group Algorithms.
Cambridge Tracts in Mathematics 152, Cambridge University Press, 2003.