On the Algebraic Structure of Convergence

Alva Couch and Yizhan Sun
Tufts University
couch@cs.tufts.edu,
ysun@cs.tufts.edu

Background

- System and network administration (network configuration management)
- CFengine provides convergent behavior.
- Observation: compositions of convergent processes are not always convergent.
- Example: file editing.

Convergent Configuration Management Challenges

- Why can compositions of convergent actions lead to confusing and even divergent behaviors?
- What limits on practice will assure predictable responses to convergent processes?

Our Approach

- Express self-healing as a result of applying sequences ***F***(P) from a finite set of convergent operations P={ p_1, p_2, … p_n }.
- While ***F***(P) is infinite, effects of ***F***(P) on a particular machine are finite.
- Express algebraic properties of ***F***(P) as **equivalence of effect**, e.g., p≈q means that p and q have the same effect.
- Study factor structure ***F***(P)/≈, a **finite** set of equivalence classes of operations.

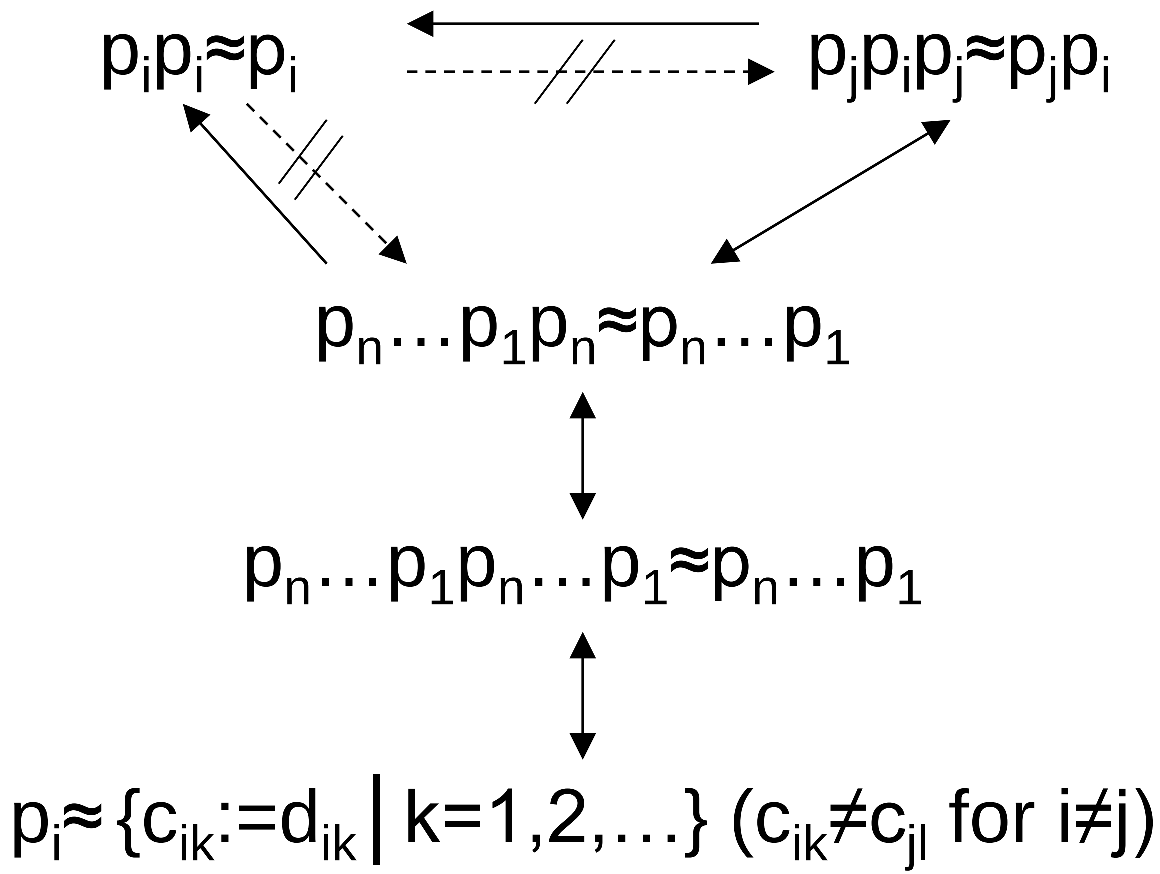
Why Equivalences are Important

- ***F***(P)/≈ (the set of equivalent classes of operations) represents **achievable states**.
- Expense of validating a self-healing system varies with the number of achievable states.

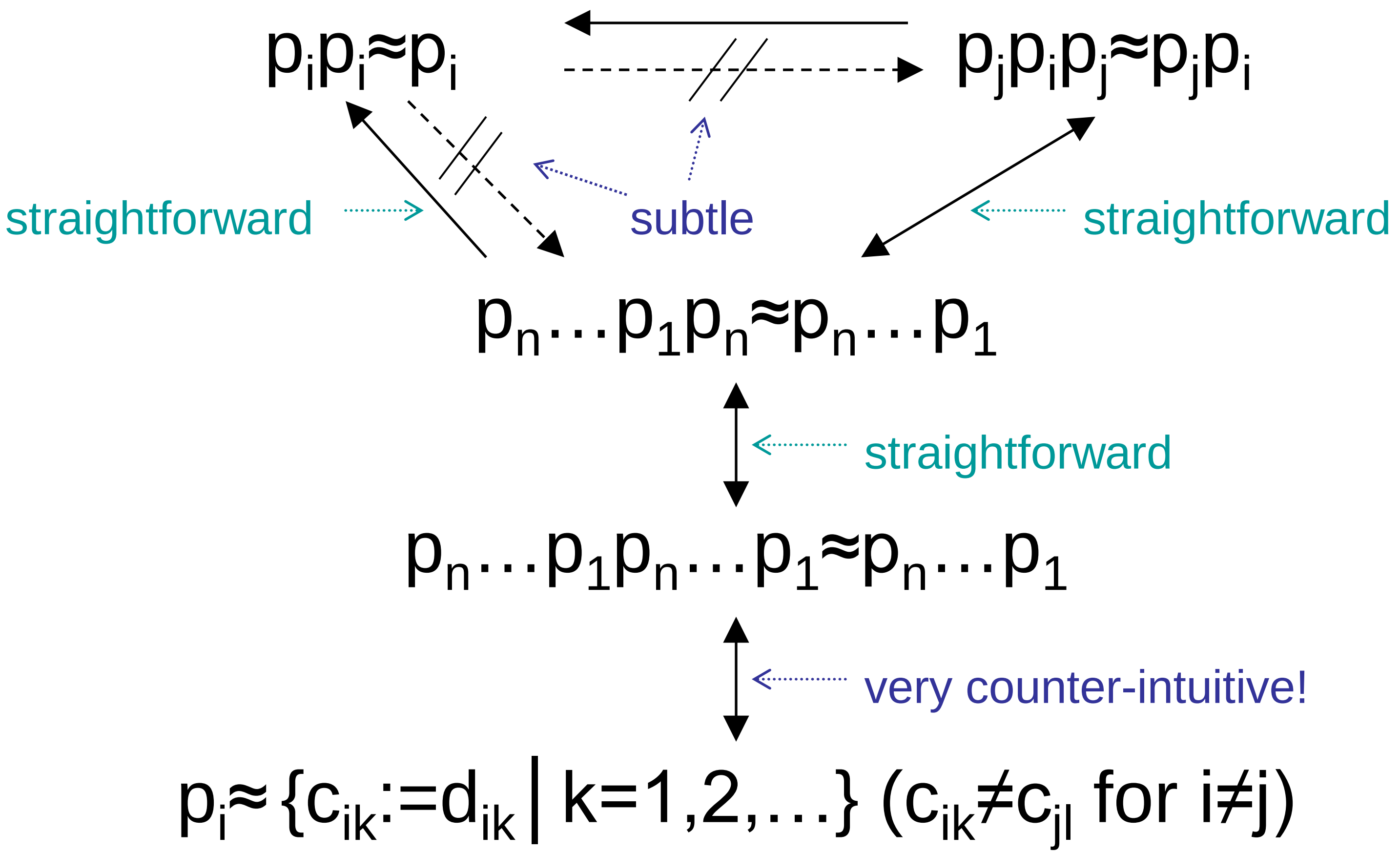
Kinds of Algebraic Equivalences

- Idempotence: $p_i p_i \approx p_i$
- Pairwise statelessness: $p_j p_i p_j \approx p_j p_i$
- Statelessness: $p_n \ldots p_1 p_n \approx p_n \ldots p_1$
- Sequence idempotence (or idempotence of ***F***(P)): $p_n \ldots p_1 p_n \ldots p_1 \approx p_n \ldots p_1$
- Operations are written right to left, i.e., $p_n \ldots p_1(S) = p_n(\ldots(p_1(S))\ldots)$

Preliminary Algebraic Results



Preliminary Algebraic Results



pp≈p and qq≈q does not insure qpqp≈qp

- Baseline: x=y=0
- p: if (x==1) then y:=2
- q: x:=1
- qp: { x=1, y=0 }
- qpqp = q(pq)p: { x=1, y=2 }
- A composition qp of idempotent actions q,p need not be idempotent.

Case Study: CFengine File Editing

```
editfiles:
  all::
  { /etc/services
    hashCommentLinesContaining “tftp”
    appendIfNotPresent “tftp 6900/udp”
  }
```

- Each operation by itself is convergent.
- Paired, they fill the file with useless comments.
- Consider what happens if one uses
  `uncommentLinesContaining “tftp”`
  on the result.

More Editing Problems

- `deleteLinesMatching “ftp”`
  - Not specific enough; will delete lines containing “tftp” as well as “ftp”.
- `appendIfNotPresent “tftp 6800/udp”`
  - Does not sense duplicate records with different port.

What Goes Wrong With Editing

- Non-convergent compositions allow proliferation of latent states.
- State proliferation causes uncertainty in applying further edits.
- Problem is syntax. Instead we need something like:

```
assert service=tftp port=6900 proto=udp
retract service=tftp
```

Statelessness

- A set of operations is **stateless** if the result of a single operation q is independent of any prior application: $qp_n \ldots p_1 q \approx qp_n \ldots p_1$
- Property of a **set** of operations, not a single operation.
- Depends upon choice of **baseline** state.
- **Sufficient** but not **necessary** to prevent state proliferation.

Facts about Statelessness

- Sufficient but not necessary to assure sequence idempotence: $p_n \ldots p_1 p_n \ldots p_1 = p_n \ldots p_1$
- Sequence idempotence has some nice properties:
  - Every sequence equivalent to one including each operation at most once
  - Resulting state space is finite with size $\leq 2^n$, n=number of operations

A Curious Result

- For stateless sets of operations, we can prove that configuration parameters exist!
- A **band** is a semigroup for which all elements are idempotent: pp=p.
- A **commutative band** is one in which pq=qp for all p,q.
- A **matrix band** is one in which pq≠qp for all p,q.

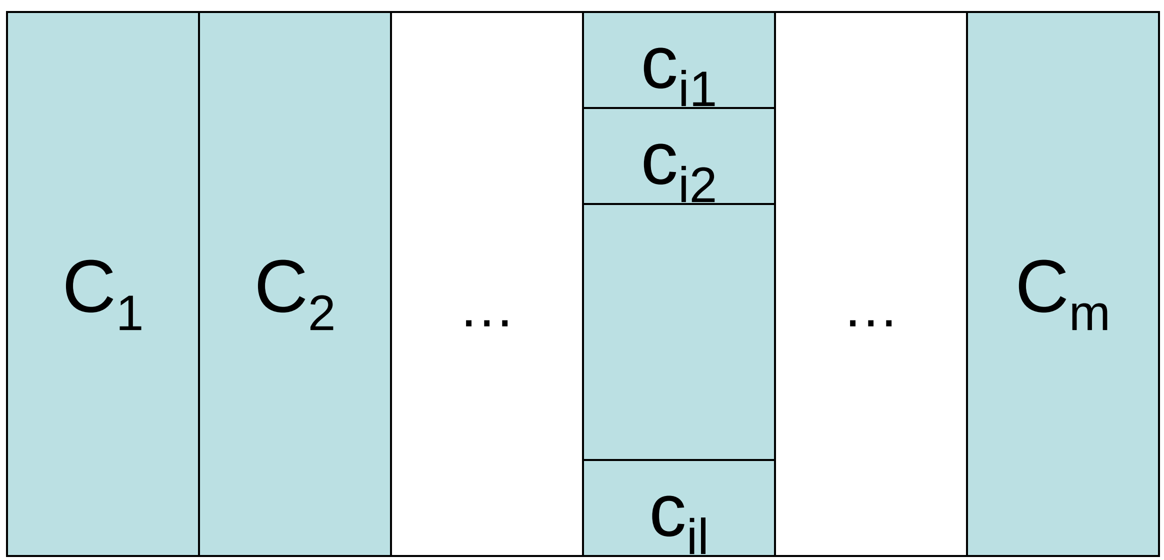
The Structure Theorem

- If P is sequence-idempotent, then ***F***(P)/≈ can be viewed as a commutative band of matrix bands of unit groups. Construction:
  - Express F(P)/≈ as a disjoint union of subsemigroups $C_{i,}$ where the C_i form a semigroup themselves.
  - Define C_jC_i as the unique set C_k where for c_i in C_i and c_j in C_j, c_jc_i is in C_k.
  - This can be done to ensure that $\{C_i\}$ is commutative, while each C_i by itself is a matrix band.

Inferred Parameters

- $C_1 \ldots C_m$ represent orthogonal **parameters** $(C_iC_j=C_jC_i)$
- Contents of each C_i represent **settings** $(c_{1i}c_{2i} \neq c_{i2}c_{i1})$

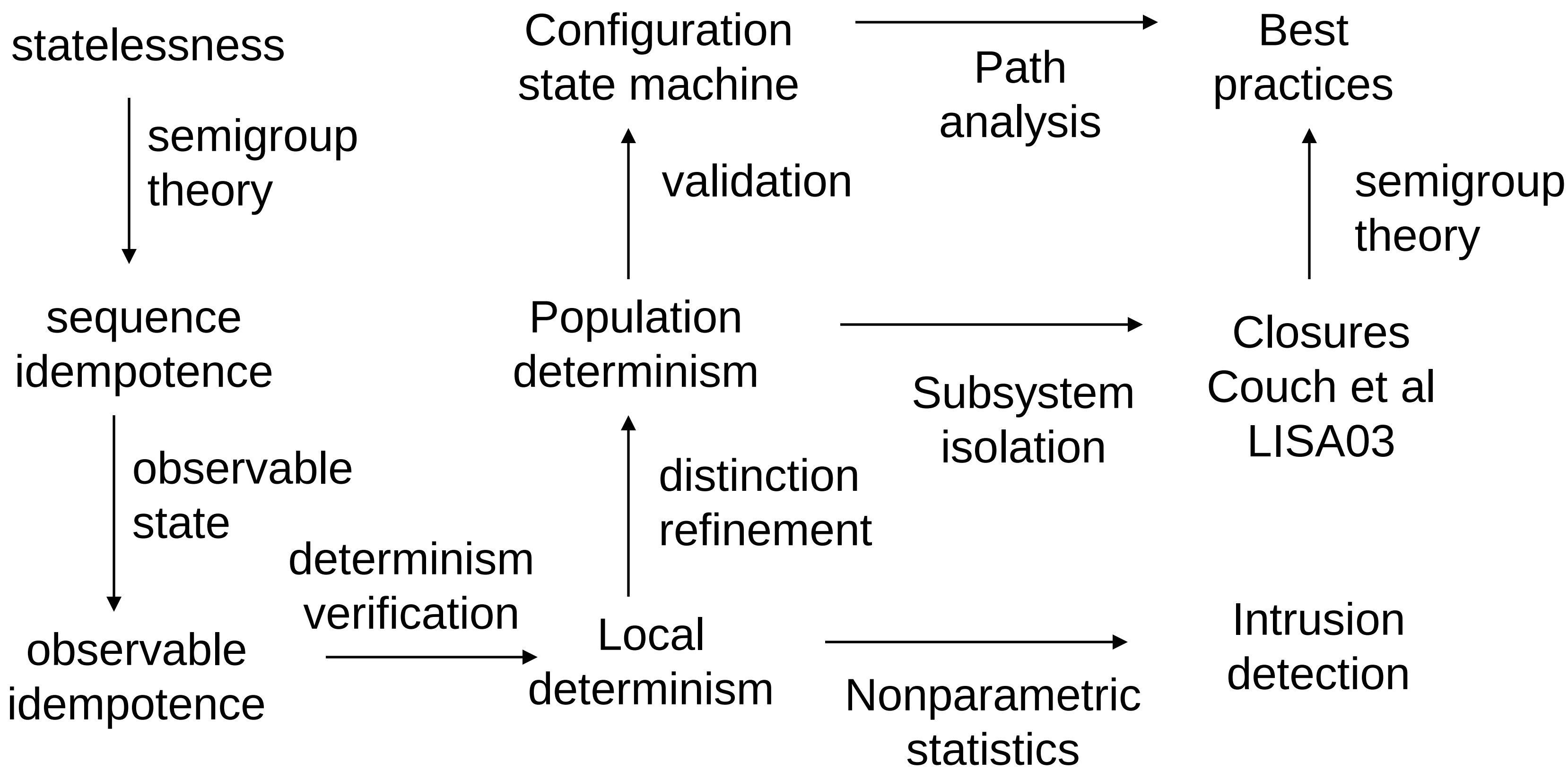
F(P)/≈



Conclusions

- Statelessness of operations leads to sequence idempotence
- Sequence idempotence is highly desirable
  - Reduction of achievable states
  - Creation of an ideal parameter space
- Achieving sequence idempotence requires changes in practice
  - Avoiding stream edits
  - Expressing changes as assertions.

Just a Beginning



More Information

- Alva L. Couch
Computer Science
Tufts University
Medford, MA USA 02155
couch@cs.tufts.edu
- LISA Paper: Couch, Hart, Greenlee, and Kallas, “Seeking Closure in an Open World”, Proc. LISA03, Oct 29-31,2003