Range-Based Set Reconciliation

Aljoscha Meyer

Set Reconciliation

- set union over a network
- between (exactly) two machines
- unstructured data
- no shared state or history

Trivial Reconciliation

Model and Analysis

- Alfie and Betty talk over a network
- reliable communication, rounds of unit length, unlimited bandwidth
- probabilistic solutions

Model and Analysis

- Alfie and Betty talk over a network
- reliable communication, rounds of unit length, unlimited bandwidth
- probabilistic solutions
- n: size of the union
- $n_{\triangle}$: size of the symmetric difference

Model and Analysis

- roundtrips
- communicated bytes
- computation time per reconciliation session
- computation space per reconciliation session
- computation time per item
- computation space per item

P2P Reconciliation

Peer-to-peer systems:

- iterating over local set infeasible
- loading local set into memory infeasible
- some peers are out to get us

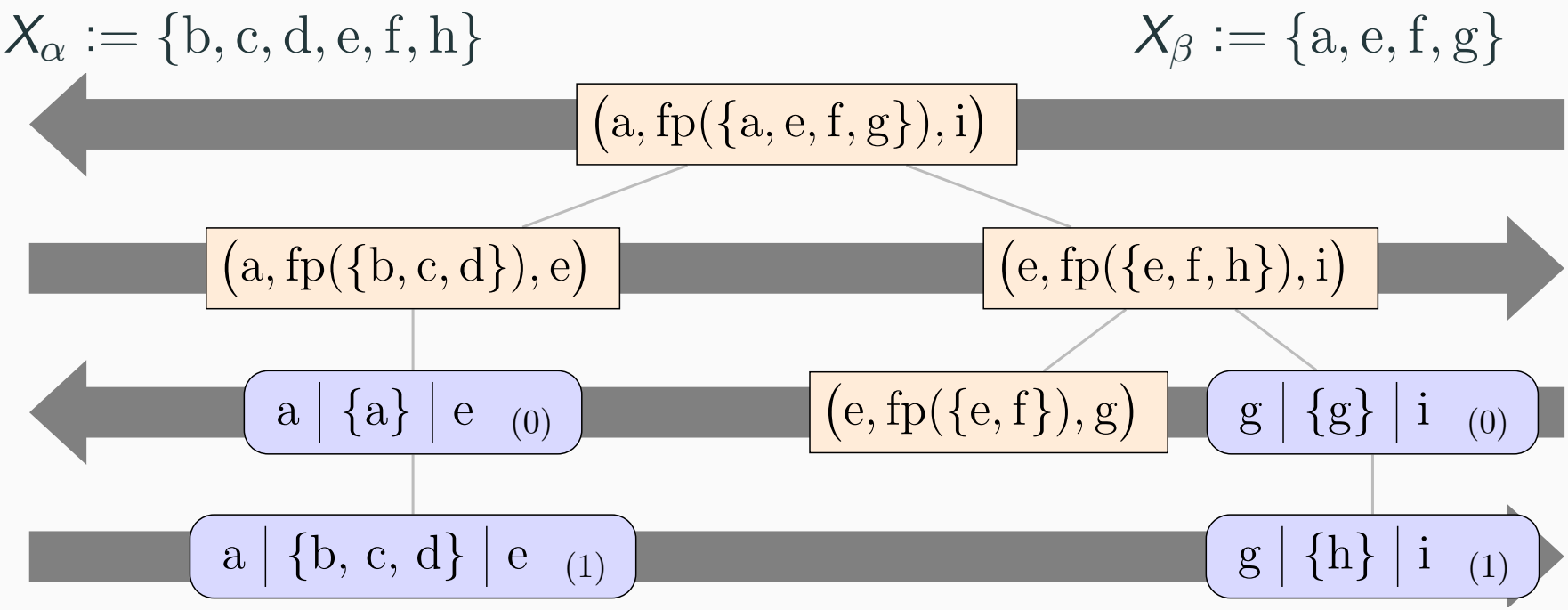
P2P Reconciliation

Peer-to-peer systems:

- iterating over local set infeasible
- loading local set into memory infeasible
- some peers are out to get us

$\Longrightarrow$ traditional approaches don't work

Range-Based Set Reconciliation



Some Nice Properties

- reasonably efficient: $\mathcal{O}(\min(n_{\triangle} \cdot \log(n), n))$ bytes communication, $\mathcal{O}(1)$ working memory
- arbitrary recursion anchor protocols
- arbitrary partition techniques

Some Nice Properties

- reasonably efficient: $\mathcal{O}(\min(n_{\triangle} \cdot \log(n), n))$ bytes communication, $\mathcal{O}(1)$ working memory
- arbitrary recursion anchor protocols
- arbitrary partition techniques
- but: linear computation times

Reducing Computation Times

- Step 1: Put a Merkle tree on it
- Step 2: ???
- Step 3: Profit

- not-so-subtle: linear computation times
- first approach (kinda-acceptable but not really): put a Merkle tree on it
- explain Merkle trees
- different trees of the same data leads to different root labels -¿ history-independent data structure
- cannot fingerprint arbitrary ranges -¿ let tree shape guide reconciliation
- MSTs
- problem 1: everyone has to use that one representation
- problem 2: adversarial trees
- blueskies...

- let's backtrack and solve the first problem -¿ associativity
- example tree (using counts, then hashes)

- definition
- arbitrary ranges: collections of maximal subtrees, complexity analysis
- successive ranges yay

- this gives us worst-case bounds on roundtrips
- implementation independence
- more datastructure-neutral formulation: monoids
- paper actually uses different data structure, also B-trees, skip lists, radix trees, no data structure, ... and all interoperable

- malicious adversaries: active vs passive
- even passive finds collisions in relatively small sets
- but: collision must actually affect a session -¿ brittle, and randomization ftw

- But what if there are many collisions? -¿ collision-resistant hash functions

- must always lift into the monoid with a secure hash function. Interesting part is the selection of the monoid
- bellare: xor, addition, multiplication, vectorized addition (lattices)
- multiset homomorphic hashing adds rsa and elliptic curves
- we are weaker: homomorphism-flavored characterization
- in particular, no need for commutativity (because we use search trees) -¿ matrix multiplication -¿ cayley hashes
- Don't know anything else, but cayley still has unnecesary structure, as we do not need inverses

- conclusion: first non-trivial reconciliation algorithm to work in resource-constrained settings and adversarial environments

- also: comparatively simple -¿ had multiple open source developers reach out to me