Panasync: Dependency Tracking among File Copies

Carlos Baquero

Victor Fonte

Paulo Sérgio Almeida

{cbm,vff,psa}@di.uminho.pt

Distributed Systems Group, DI - U. Minho

http://gsd.di.uminho.pt/panasync/

September, 2000

Goals

The goal of Panasync is to support *ad hoc* file replication, by tracking causality among an unbounded number of autonomous replicas.

The target scenario assumes:

- Very loose connectivity – even only transportable media.
- No need for central servers or distributed configurations.
- No use of system clocks or real time information – only causality.

The only requirement is file system read-write access to one or two replicas at a time.

Motivation: Nomadic Trend

- Leads to increased difficulty of replica/version synchronization control.
- Can encompass relocation of nodes, by nomadic behavior.
- Stimulates spontaneous cooperation among nodes, possibly in isolated *ad hoc* connections.
- Raises opportunities for interactions using transportable media (Disks, Zips, ...).

Services: Detection

Without using semantic information on updates, the system can determine:

- If two files are related (lineage) and, if so, detect:
  - If they are equivalent.
  - If one has seen more effective updates than the other (MD5).
  - If they hold concurrent updates.

File names do not matter for replica accounting (lineage), only history of creation is important.

Services: Manipulation

At any time (with RW access to replica info) it is possible to:

- Create a new lineage of replicas.
- Duplicate a file replica.
- Merge or synchronize two file replicas.
- Rename and move files.

In fact, *almost* all the flexibility of the standard **copy** command

The path to Panasync

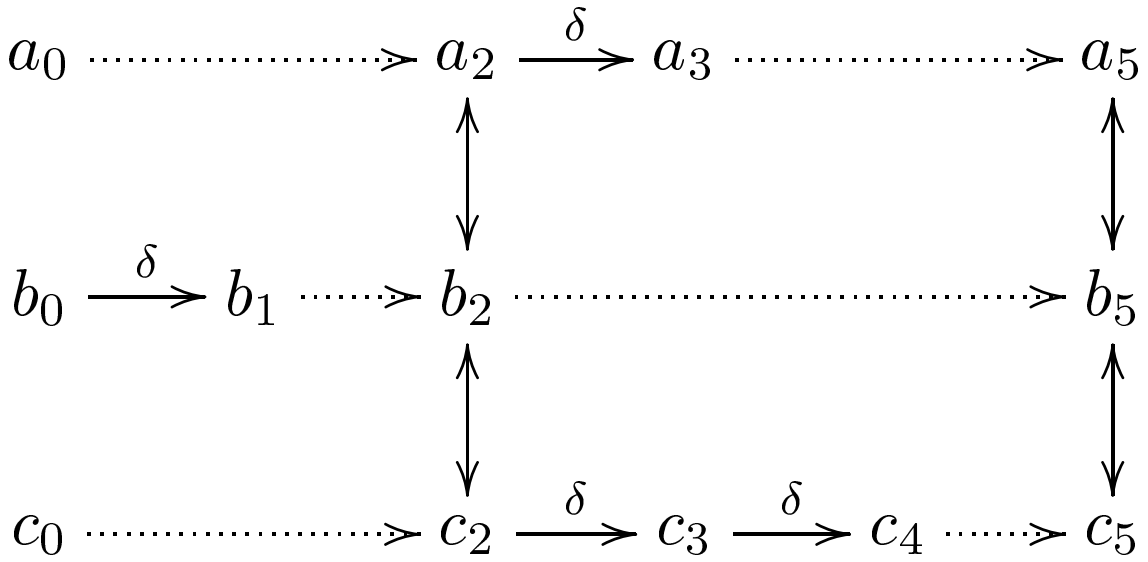
Replica control has been tied to version vectors (closely related to Lamport's causality and vector clocks).

Version vectors, as well as Lamport's causality, need a **fixed number** of replication points or a **universal naming** of those points.

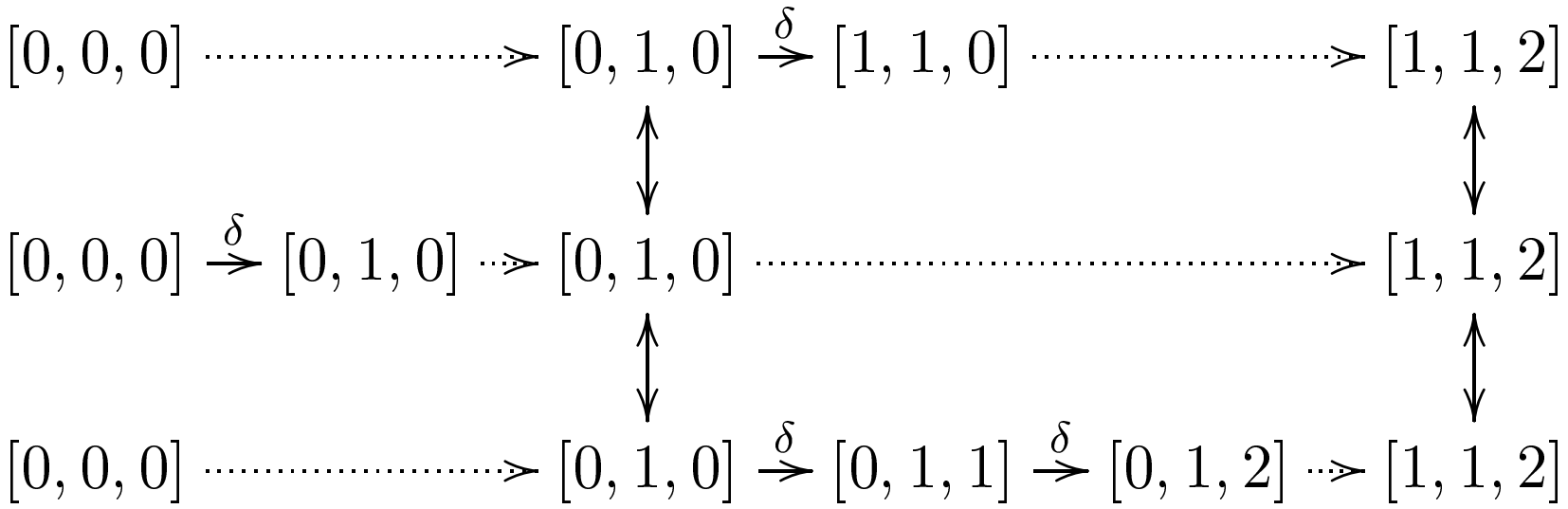
Unbounded creation and convergence of replicas motivated us to the study of a new **causality model** and **time-stamping mechanism** for replica control.

Autonomous Causality and *Autonomous Versioning*

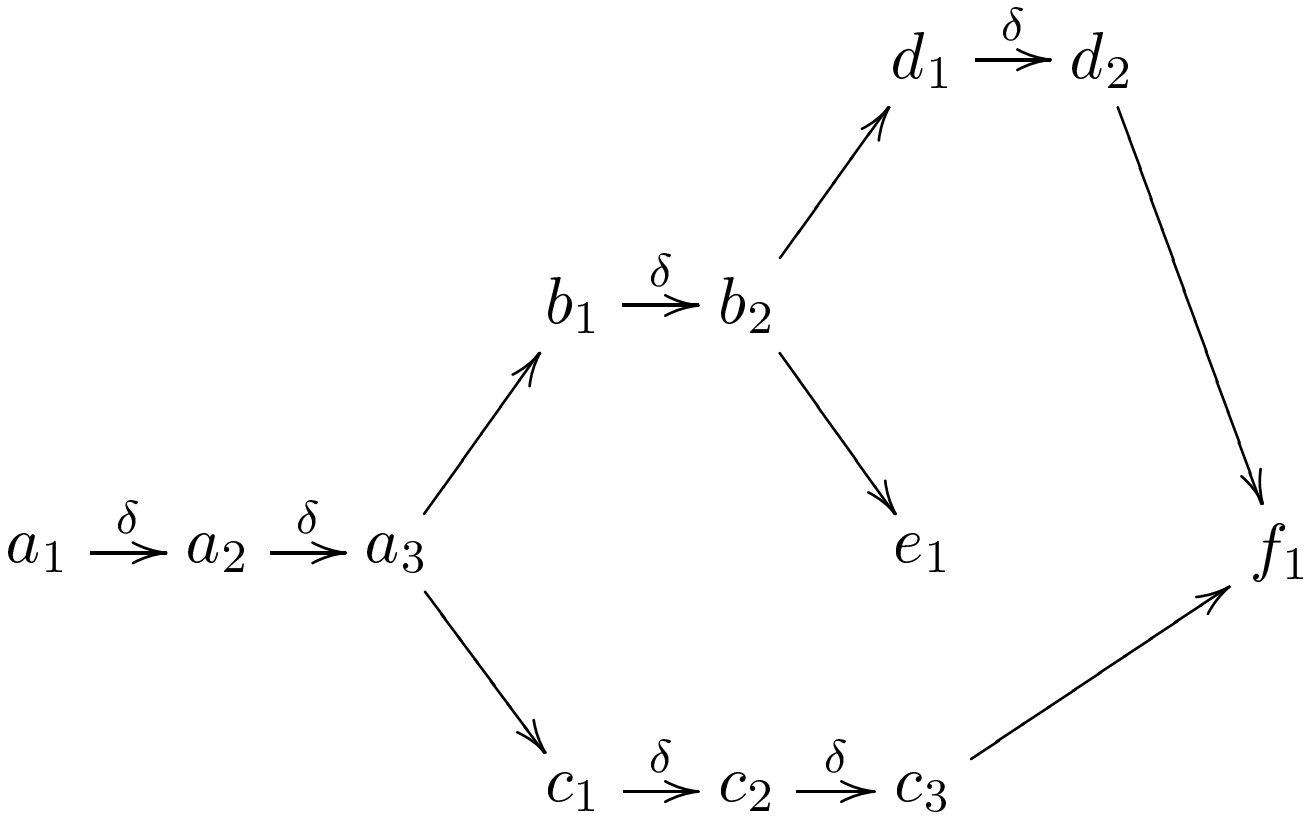
Bounded Dependencies – Version Vectors



Version Vectors

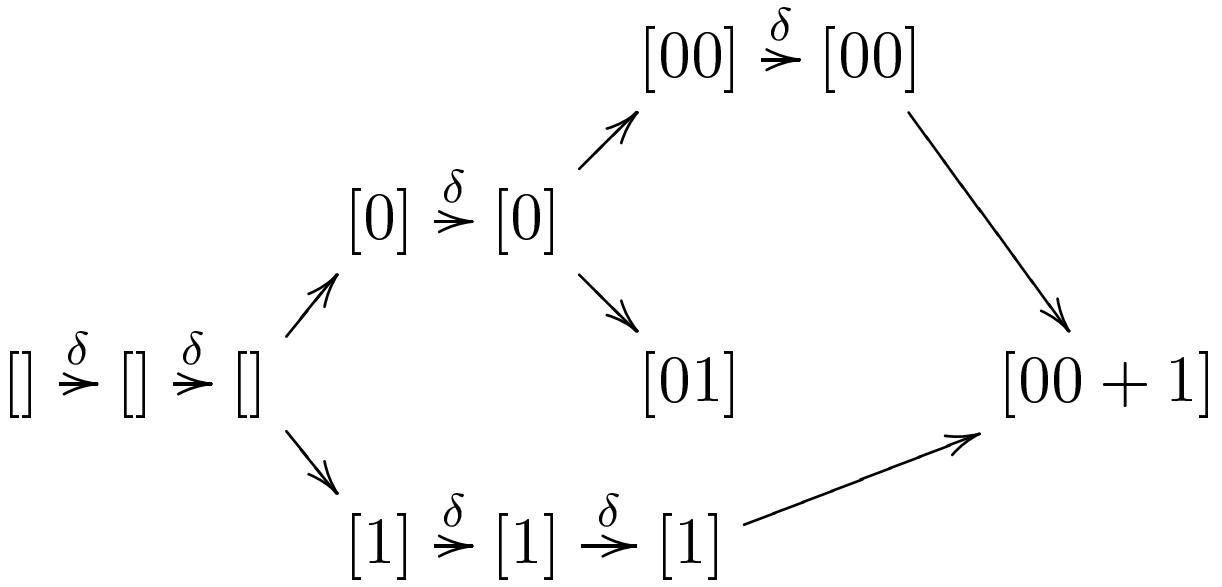


Arbitrary Dependencies – *Autonomous Causality*

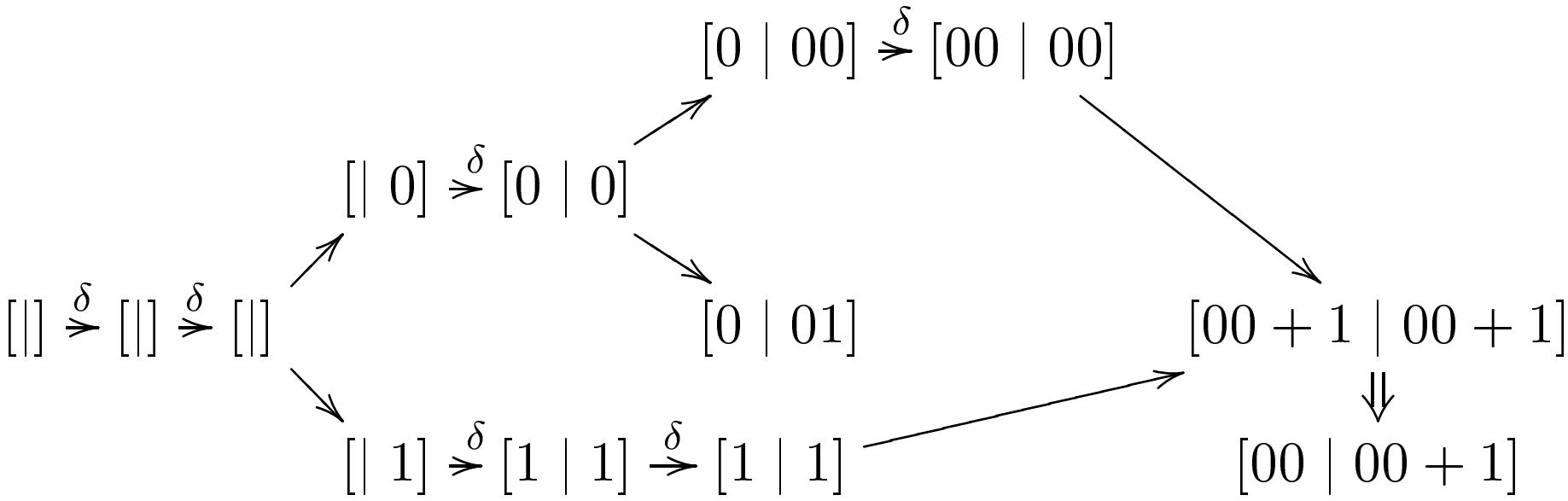


Space Partitions

Autonomous Time-Stamps (almost)



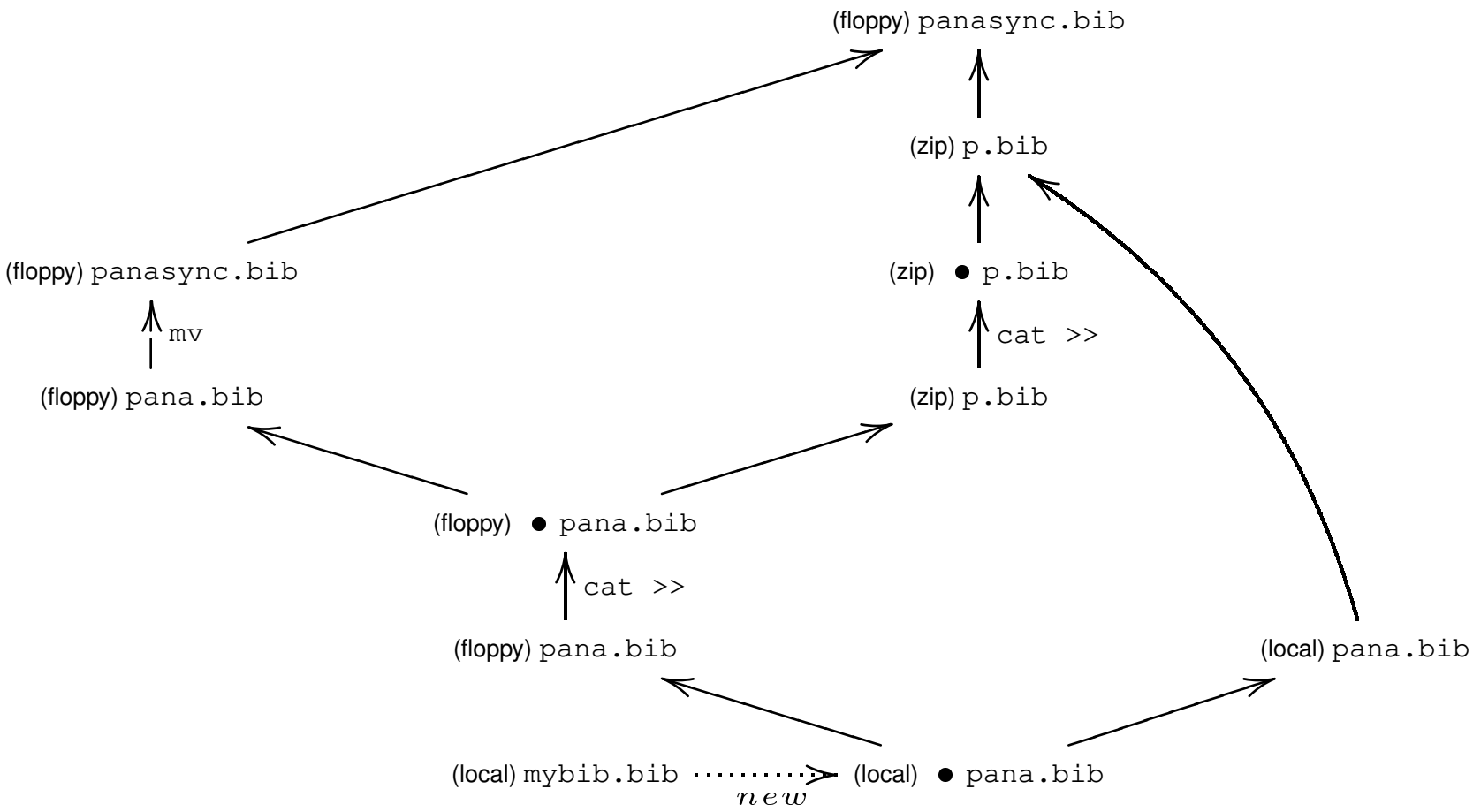
Autonomous Time-Stamps



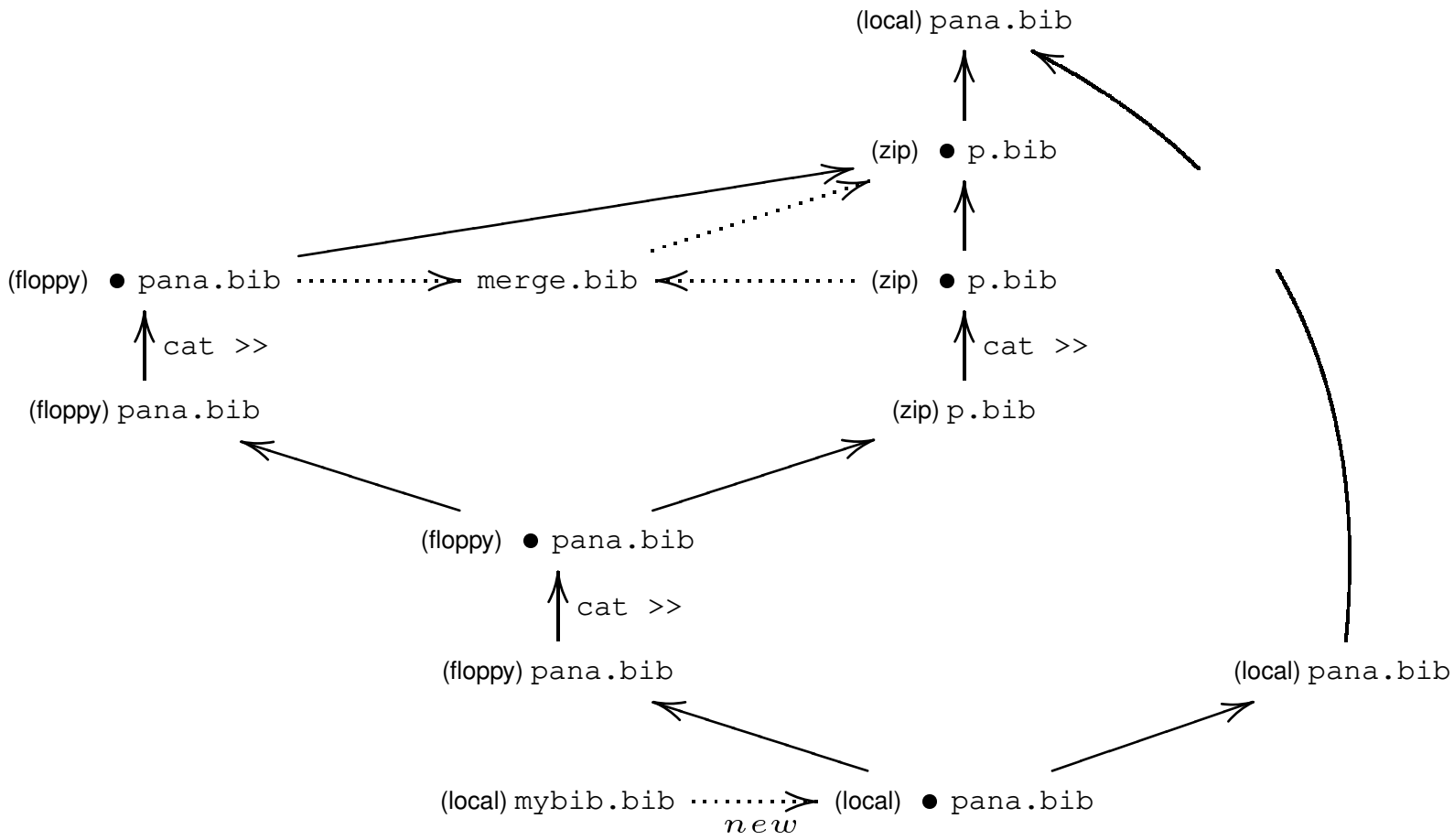
Panasync Constructs

- Create a new lineage: `pananew [base] target`
- Duplicate a p-file: `panadup base target`
- Try to join two p-files: `panajoin base target`
- Join two concurrent p-files: `panajoin base target -s content`
- Synchronize two p-files: `panasync base target [-s content]`
- List and/or compare files: `panalist [-c file]`

First Scenario: Domination



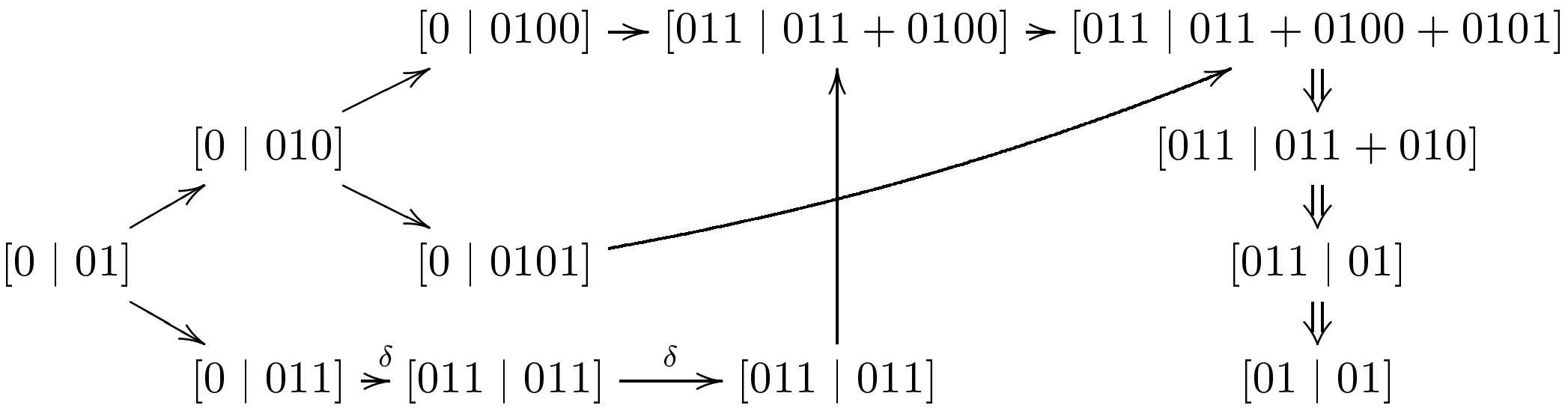
Second Scenario: Concurrency



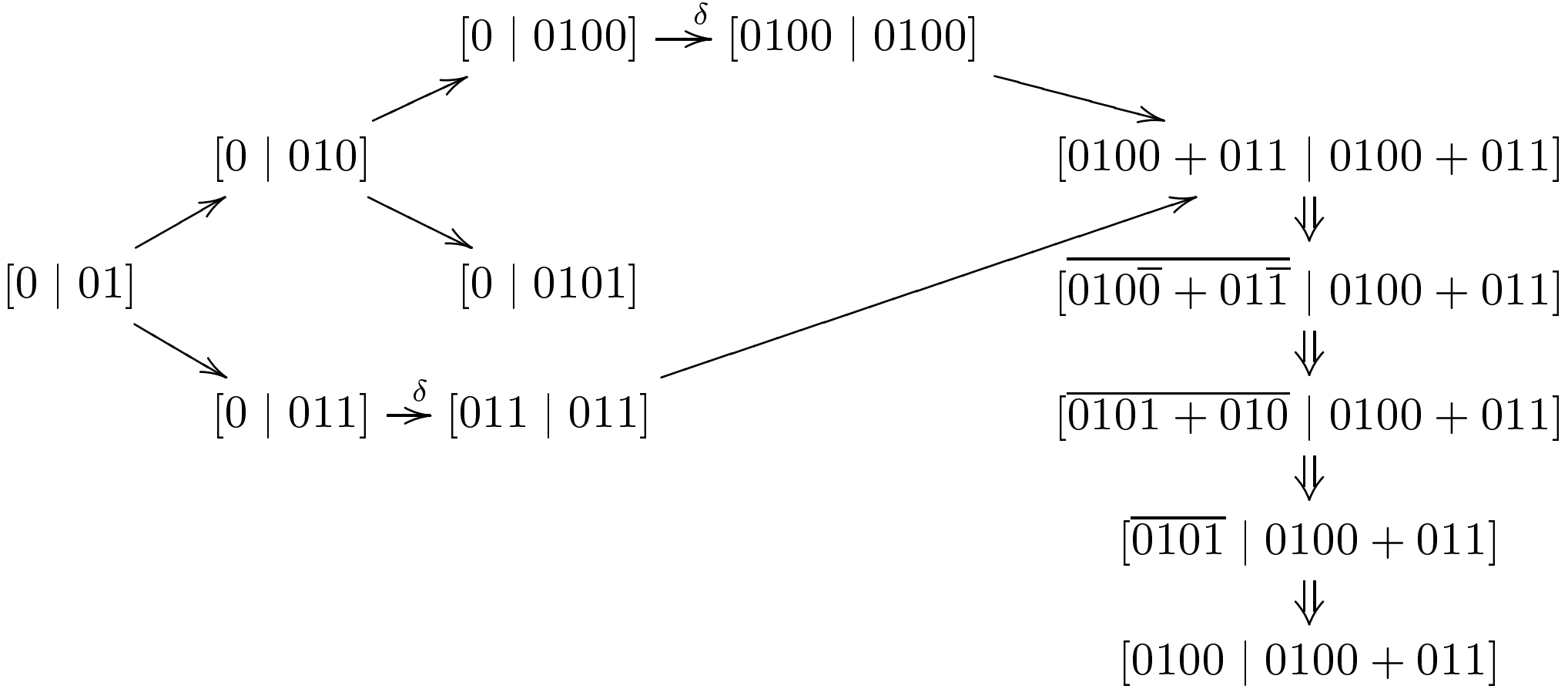
Back to Time-Stamps

- Convergence
- Simplification of First Components
- Branching after Joins

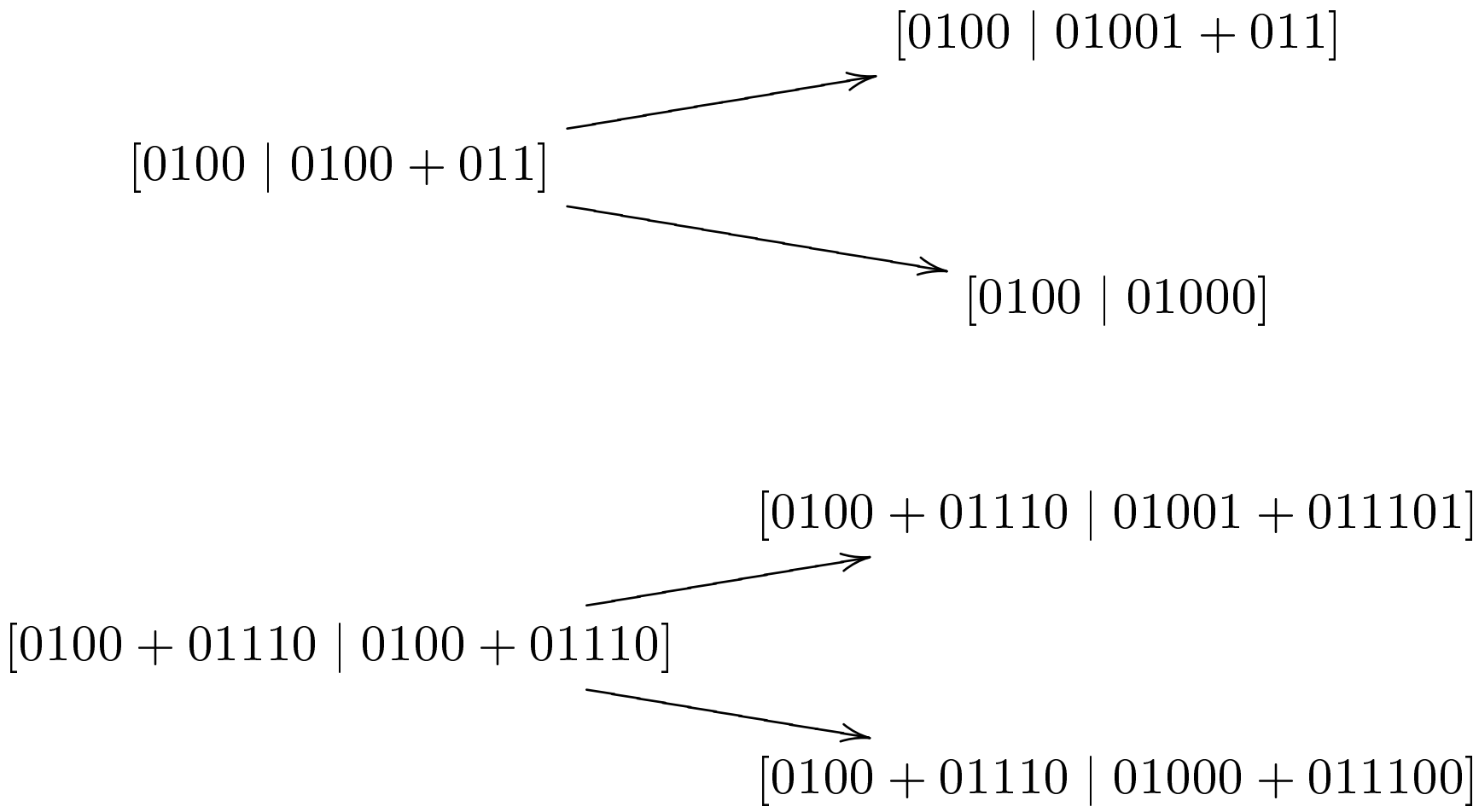
Convergence



Simplification of First Components



Branching after Joins



Weak Points and Future Issues

The current user level implementation opens some problems: The TS directory files can be deleted; Panasync files can still be subject to `mv` or `cp`; Update detection with MD5 is only statistically correct.

These problems can eventually be circunvented with file system level implementations (and file browsers).

Some future issues are: Lineage retirement; Support for deletion with no convergence.

And also: Release of a formal proof that substitutes the present consistency evaluation with causal histories.

Best Features

- Decentralization and no network connectivity requirements.
- Changes, Replication and Convergence at all time.
- Replicas resulting from joined replicas (concurrent or not) inherit their domination relation to other replicas.
- Replica lineage is independent from file names.
- Time-Stamps are dynamic, they grow and shrink.
- Total independence from systems' clocks.

Implementation

The first version of Panasync has been implemented at user level in C++ with extensive use of the STL template library, and was compiled in Linux platforms.

The Time-Stamping algorithms are provided on a separate library under LGPL, and the Panasync tools are provided as standard GPL.

`http://gsd.di.uminho.pt/panasync/`

We are open to contributions and colaboration.