Universität St.Gallen

"From insight to impact"

Local-First Software: Collaborative Spreadsheet Editing

Sandro Gössi & Fabian Gubler
Supervisors: George Zakhour & Dr. Pascal Weisenburger

Autumn Semester 2023: Integrative Master's Projects – Programming Group

University of St.Gallen

Our Team

Sandro Gössi

Master of Computer Science

University of St. Gallen

Fabian Gubler

Master of Computer Science

University of St. Gallen

Our Supervisors

George Zakhour

PhD Student, Programming Group

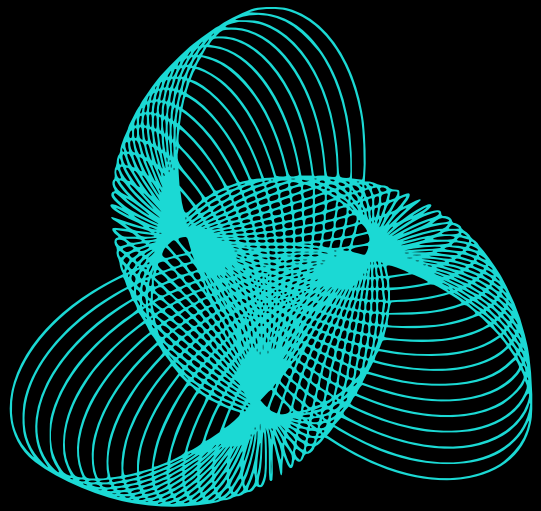
University of St. Gallen
Alumni of EPFL and AUB

Dr. Pascal Weisenburger

Postdoc, Programming Group

University of St. Gallen
Alumni of TU Darmstadt

Table Of Contents



Cellster

Promotional Pitch Video

Why Do You As A User NEED Our Product?

Motivation & Project Goal

What Was The Motivation & Goal Of Our Project?


X
Goldman
Sachs

Merging Conflict Example (Status Quo/Baseline)

B	C	D	E	F
Discounted Cash Flow Valuation Model (DCF)				
Weighted Average Cost of Capital (WACC)				
Projected Free Cash Flow (FCF)				
WACC	10%			
FCF (Year 1)	160			
FCF (Year 2)	200			
FCF (Year 3)	250			
DCF	=160/(1+0.1)+200/(1+0.1)+250/(1+0.1)			

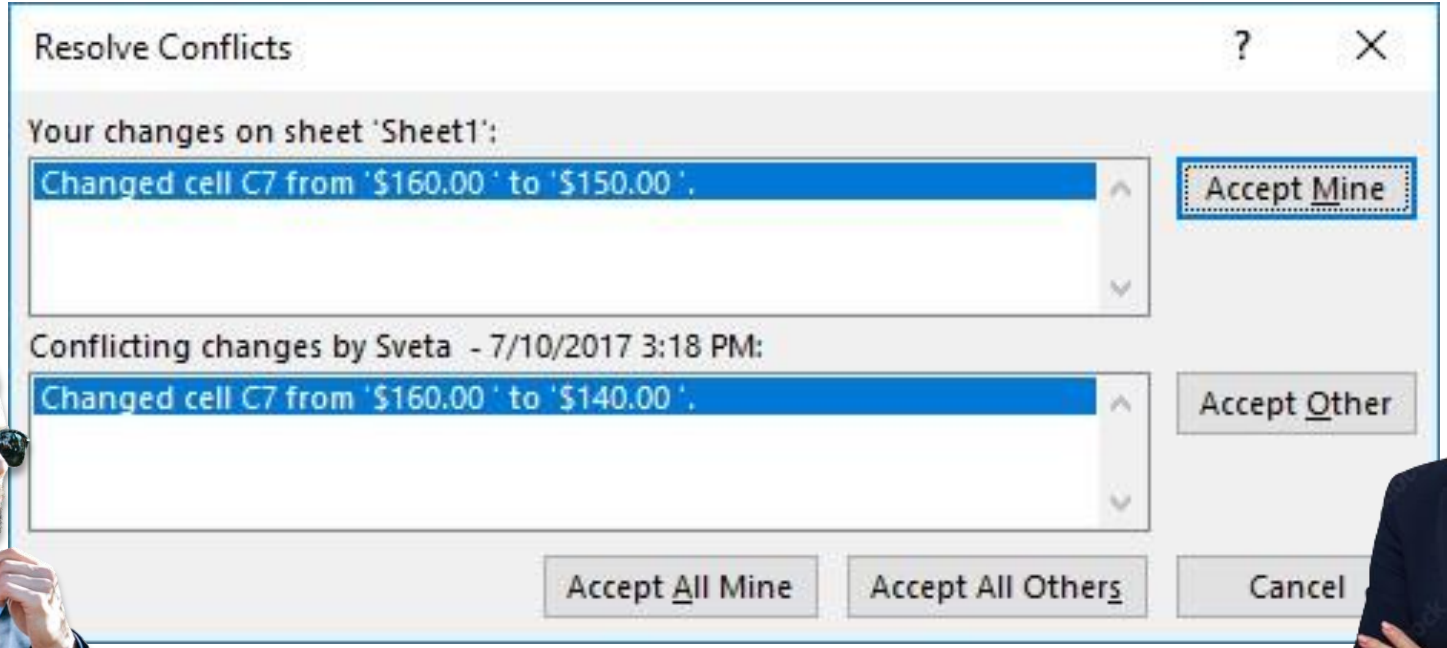
Merging Conflict Example (Status Quo/Baseline)

B	C	D	E
Discounted Cash Flow Valuation Model (DCF)			
Weighted Average Cost of Capital (WACC)			
Projected Free Cash Flow (FCF)			
WACC	10%		
FCF (Year 1)	140		
FCF (Year 2)	200		
FCF (Year 3)	250		
DCF	=140/(1+0.1)+200/(1+0.1)+250/(1+0.1)		

Merging Conflict Example (Status Quo/Baseline)

B	C	D	E
Discounted Cash Flow Valuation Model (DCF)			
Weighted Average Cost of Capital (WACC)			
Projected Free Cash Flow (FCF)			
WACC	10%		
FCF (Year 1)	150		
FCF (Year 2)	200		
FCF (Year 3)	250		
DCF	=150/(1+0.1)+200/(1+0.1)+250/(1+0.1)		

Merging Conflict Example (Status Quo/Baseline)



Merging Conflict Example (Status Quo/Baseline)

Excel 365

Simple Text Merging Does Not Work!

=150/(1+0.1)+140/(1+0.1)+200/(1+0.1)+250/(1+0.1)

=150140/(1+0.1)+200/(1+0.1)+250/(1+0.1)

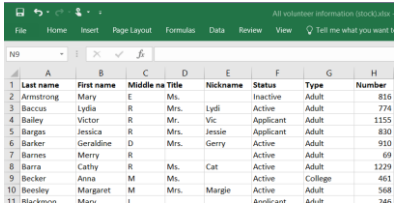
Merging Conflict Example (Status Quo/Baseline)

	User 1	User 2
Before merge	SUM(140(1+0.1))	140(1+0.1) + 50
After merge	140(1+0.1) + 50	140(1+0.1) + 50

Simple Text Merging Does Not Work!

Collaborative Local-First Spreadsheet

Project Goal: “Excel formula CRDT” – Developing A CRDT for Merging Abstract Syntax Trees of Excel Formulas/Cells



	A	B	C	D	E	F	G	H
1	Last name	First name	Middle na	Title	Nickname	Status	Type	Number
2	Armstrong	Mary	E	Ms.		Inactive	Adult	816
3	Baccus	Lydia	R	Mrs.	Lydi	Active	Adult	774
4	Bailey	Victor	R	Mr.	Vic	Applicant	Adult	1155
5	Bargas	Jessica	R	Mrs.	Jessie	Applicant	Adult	830
6	Barker	Geraldine	D	Mrs.	Gerry	Active	Adult	910
7	Barnes	Merry	R			Active	Adult	69
8	Barra	Cathy	R	Ms.	Cat	Active	Adult	1229
9	Becker	Anna	M	Ms.		Active	College	461
10	Beesley	Margaret	M	Mrs.	Margie	Active	Adult	568

Problem Statement

What Was The Concrete Problem We Wanted To Solve With Our Project?

Building The Functionality To Merge Two Excel Formulas

Excel Formula

Formula before any edits were made:

=SUM(A2:A9)

User 1

Changes the Formula to:

=SUM(A3:A10)

User 2

Changes the Formula to:

=SUM(A2:A9) / B5

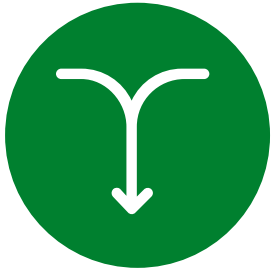
RESULT?

How do we merge this two formulas?

High Difficulty Of The Problem

Formula Complexity

Variability & Ambiguity Of Merging Rules

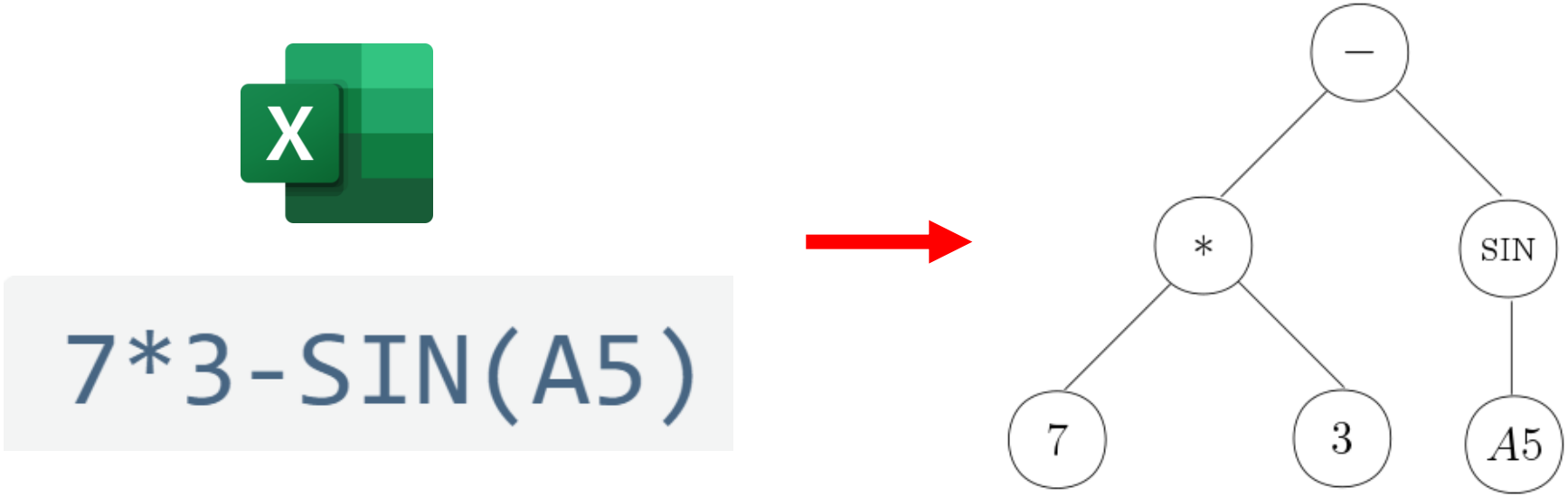


Conflict Resolution

Methodological Approach

How Did We Approach Solving The Problem?

Step 1: Parsing Formulas: Abstract Syntax Tree (AST)




```
[8]  formula = 'SUM(ABS(A1:A100)) + IF(true, INIT_VALUE, B12)*4' # String
     parse (formula) # Abstract Syntax Tree (AST)
```

```
Binary[+][Func[SUM][1][Func[ABS][1][Range[Cell[A1]][Cell[A100]]]]][Binary[*][Func[IF][3][Bool[True], Name[INIT_VALUE], Cell[B12]]][Num[4.000000]]]
```

So Let's Start And Parse Our Example!

Excel Formula
Formula before any edits were made:

=SUM(A2:A9)

User 1
Changes the Formula to:

=SUM(A3:A10)

User 2
Changes the Formula to:

=SUM(A2:A9) / B5

RESULT?
How do we merge this two formulas?

Excel Formula

Formula before any edits were made:

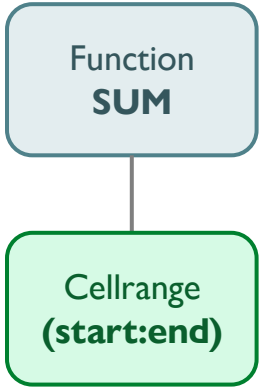
=SUM(A2:A9)

Function
SUM

Excel Formula

Formula before any edits were made:

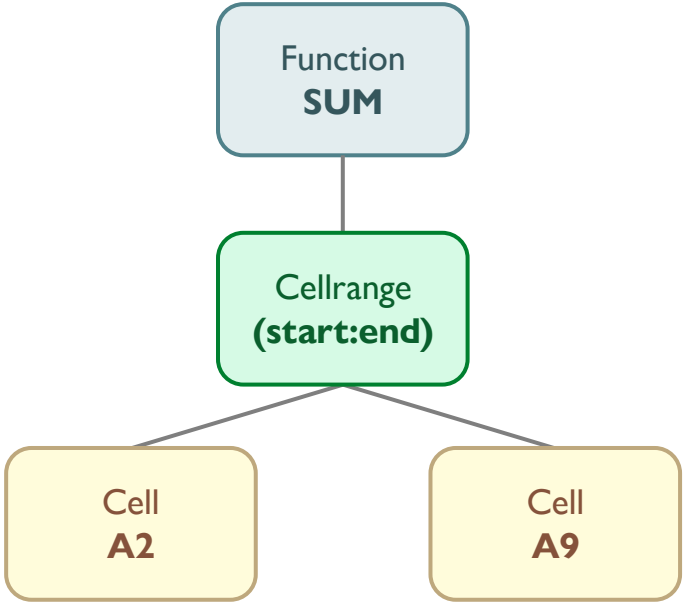
=SUM(A2:A9)



Excel Formula

Formula before any edits were made:

=SUM(A2:A9)



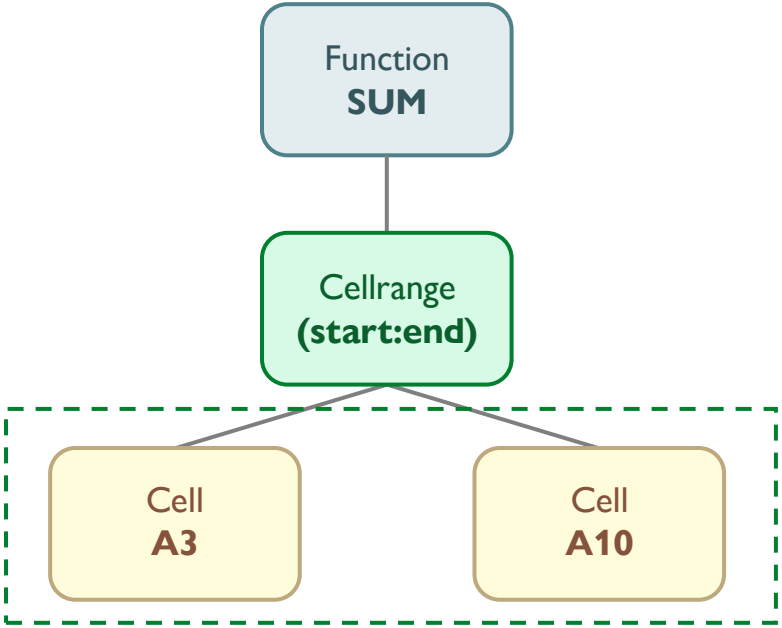
User 1 changed two “Cell Nodes”

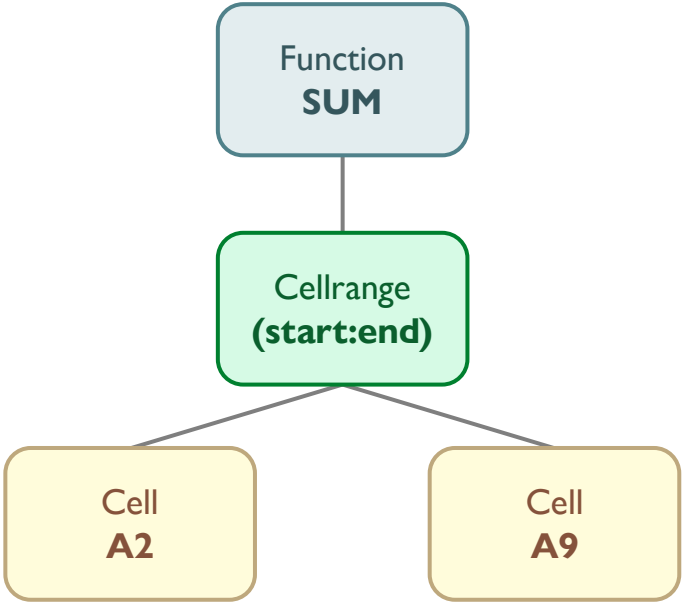
Excel Formula
Formula before any edits were made:

=SUM(A2:A9)

User 1
Changes the Formula to:

=SUM(A3:A10)



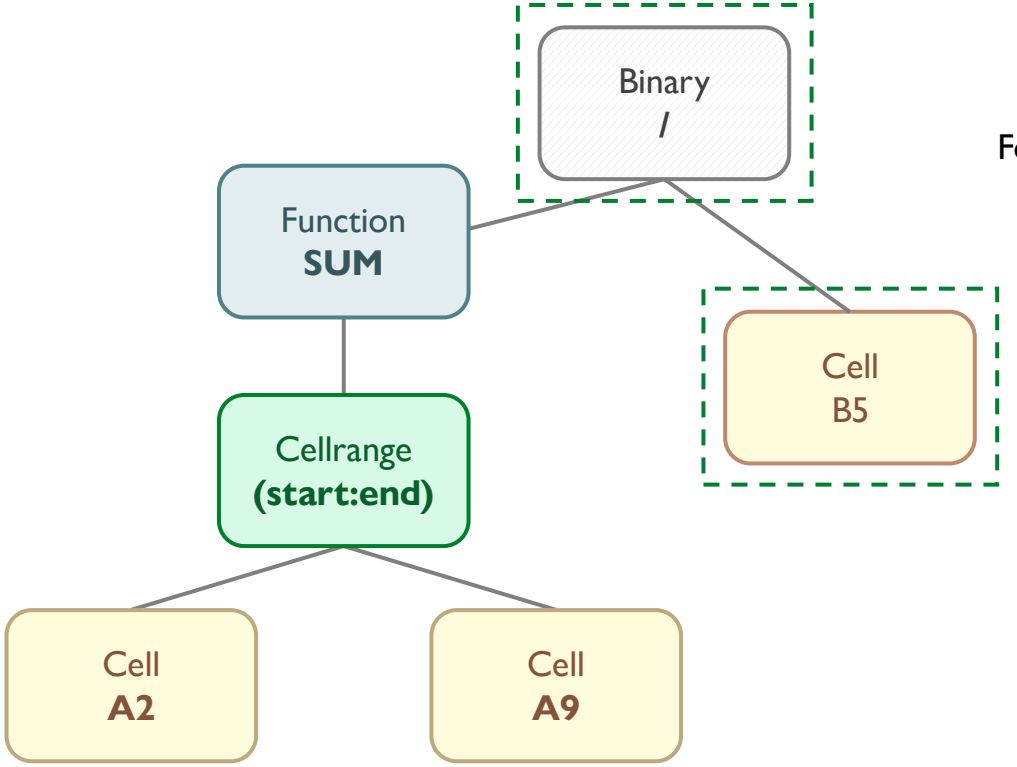


Excel Formula

Formula before any edits were made:

=SUM(A2:A9)

User 2 added two Nodes



Excel Formula

Formula before any edits were made:

=SUM(A2:A9)

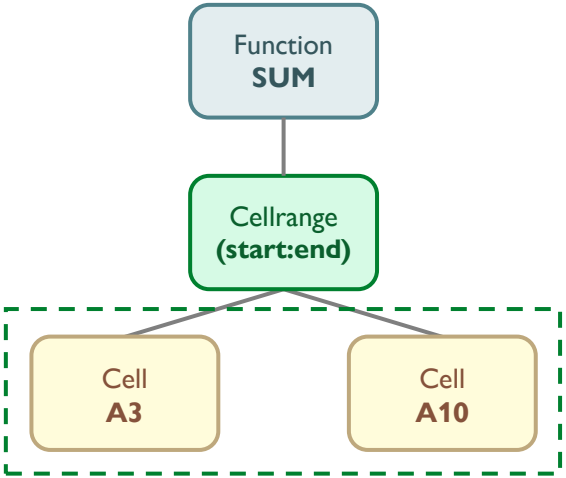
User 2

Changes the Formula to:

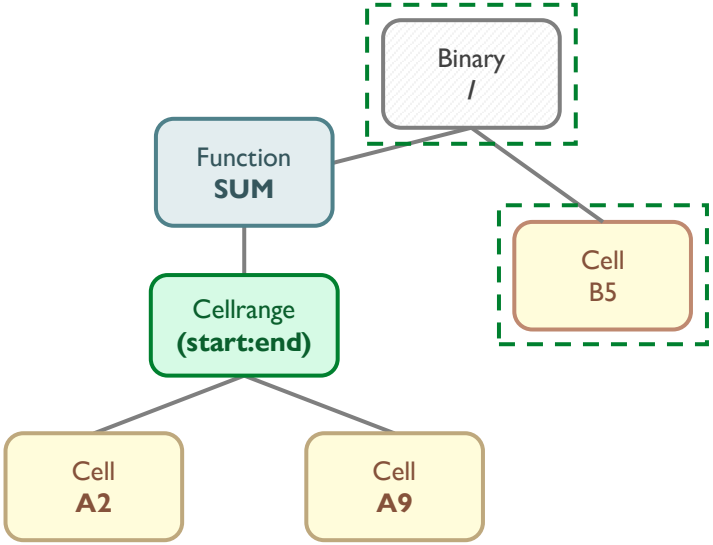
=SUM(A2:A9) / B5

How do we merge these two ASTS?

User 1



User 2

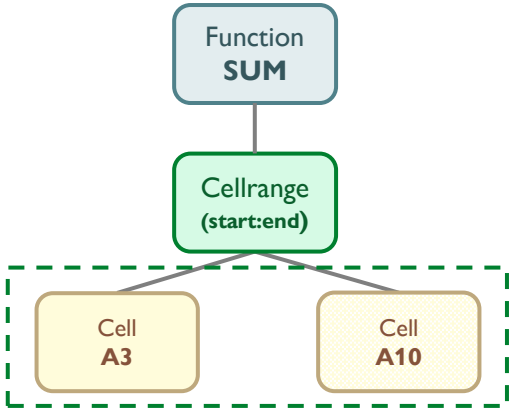


First Focus

Approach 1: Rule-Based Method

Approach 2: History-Based Method

User 1

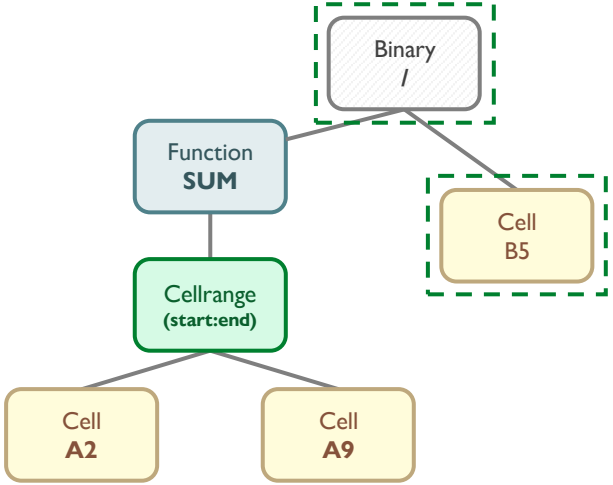


Merge Function

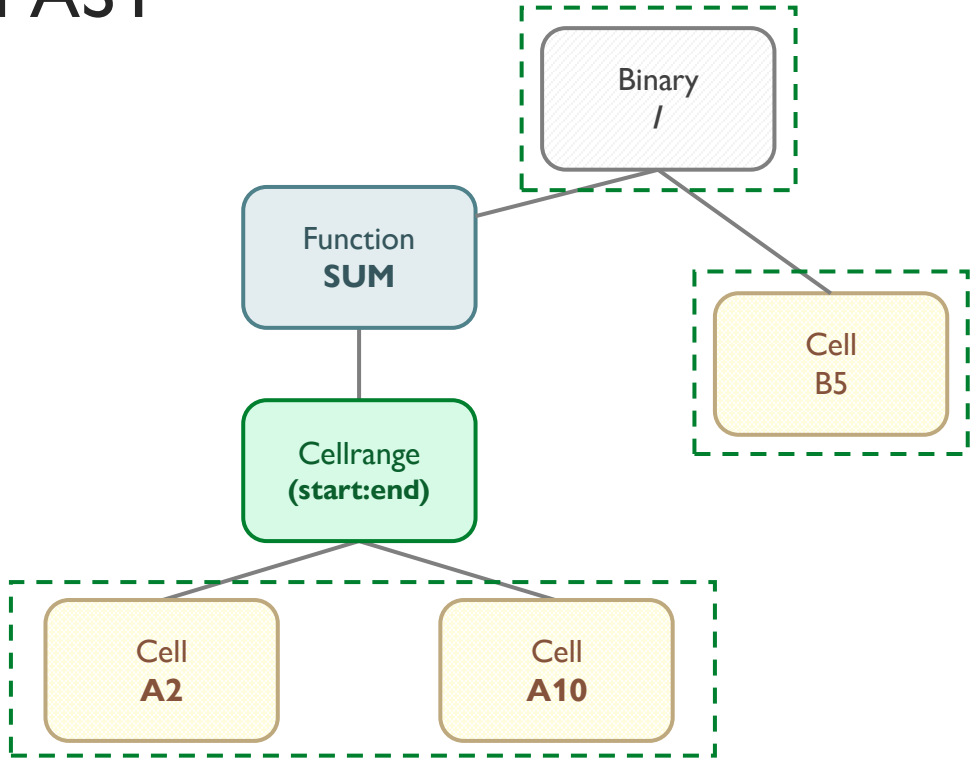
Sends complete State
(to User 1)

Sends complete State
(to User 2)

User 2



Merged AST



- **Idempotent**
- **Commutative**
- **Associative**

Approach 1:
Rule-Based
Method

Iterative Process...

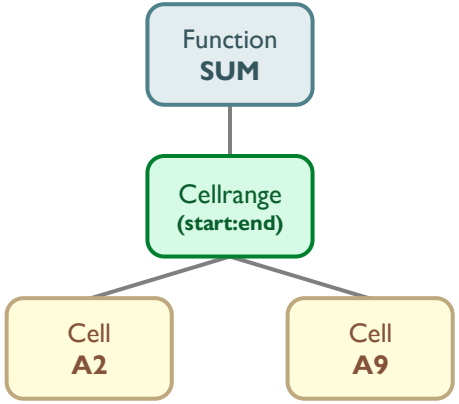
Merging Rule → Test Cases → Merge Function

Approach 1:
Rule-Based Method

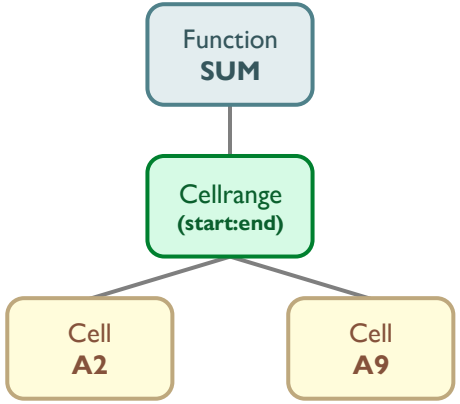
Second Focus

Approach 2:
History-Based Method

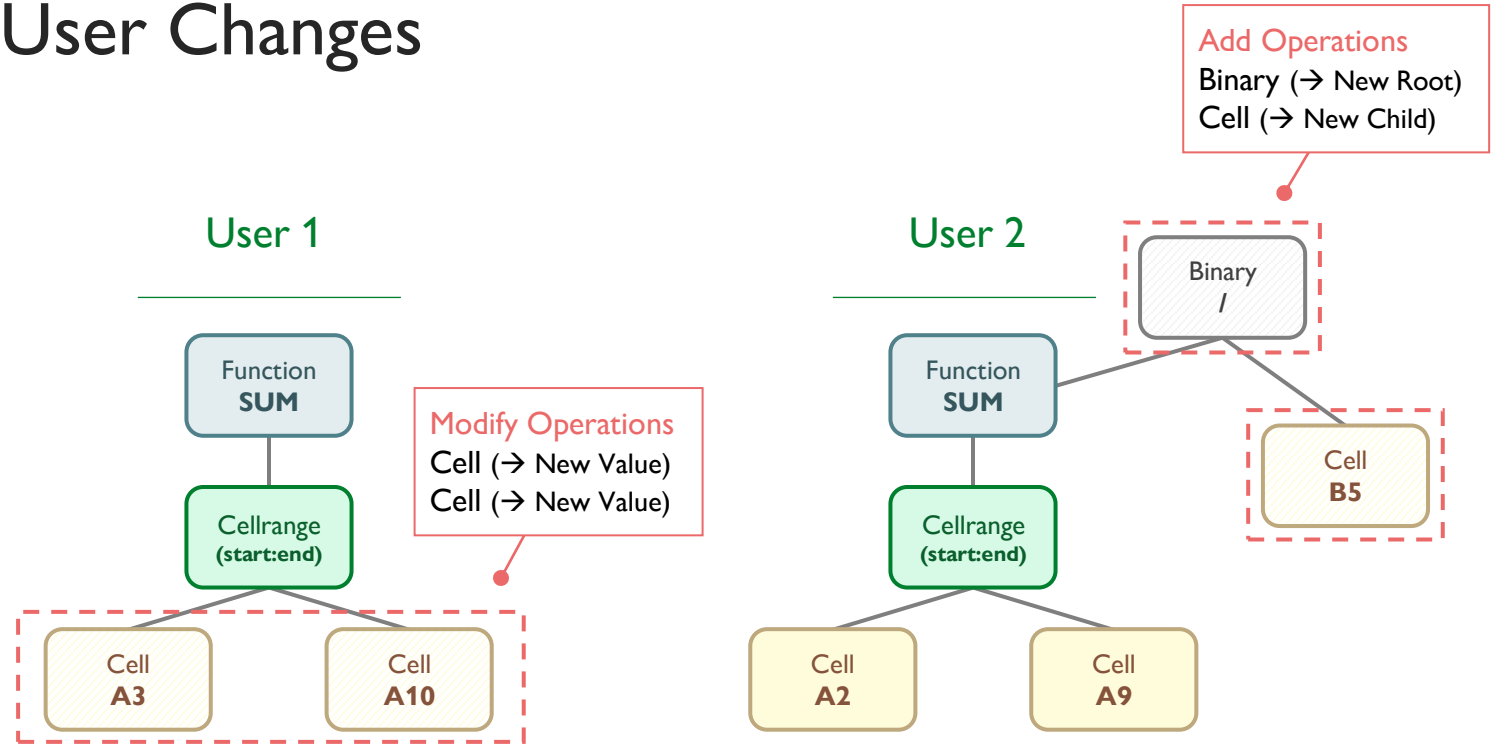
User 1


Function
SUM
Cellrange
(start:end)
Cell
A2
Cell
A9

User 2


Function
SUM
Cellrange
(start:end)
Cell
A2
Cell
A9

Isolate User Changes



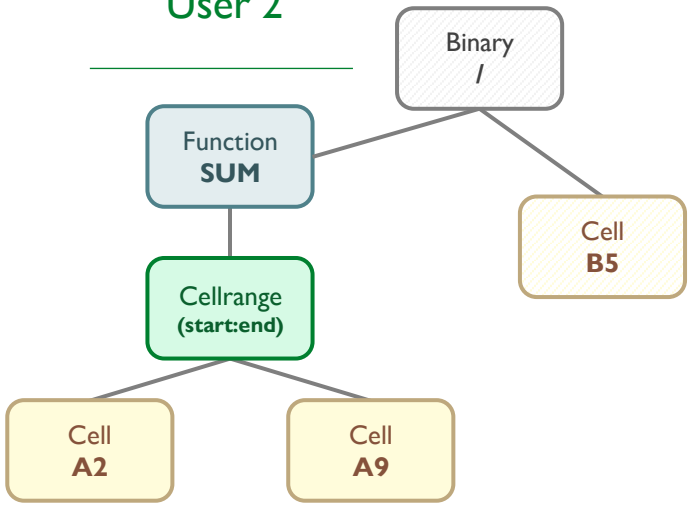
Apply User Changes

User 1

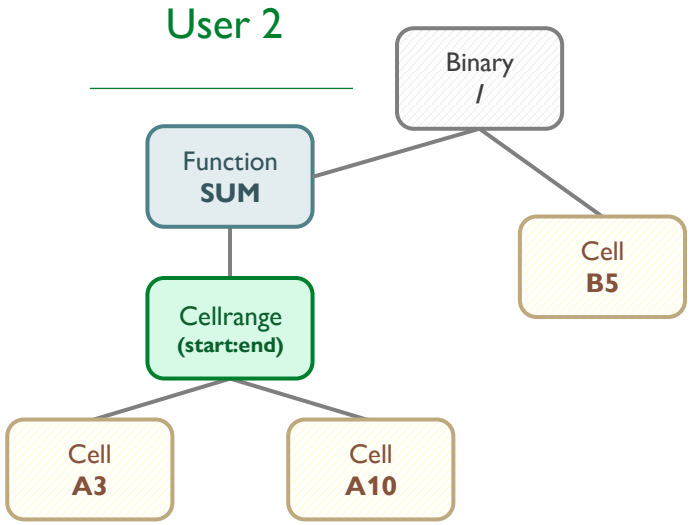
Cell
A3

Cell
A10

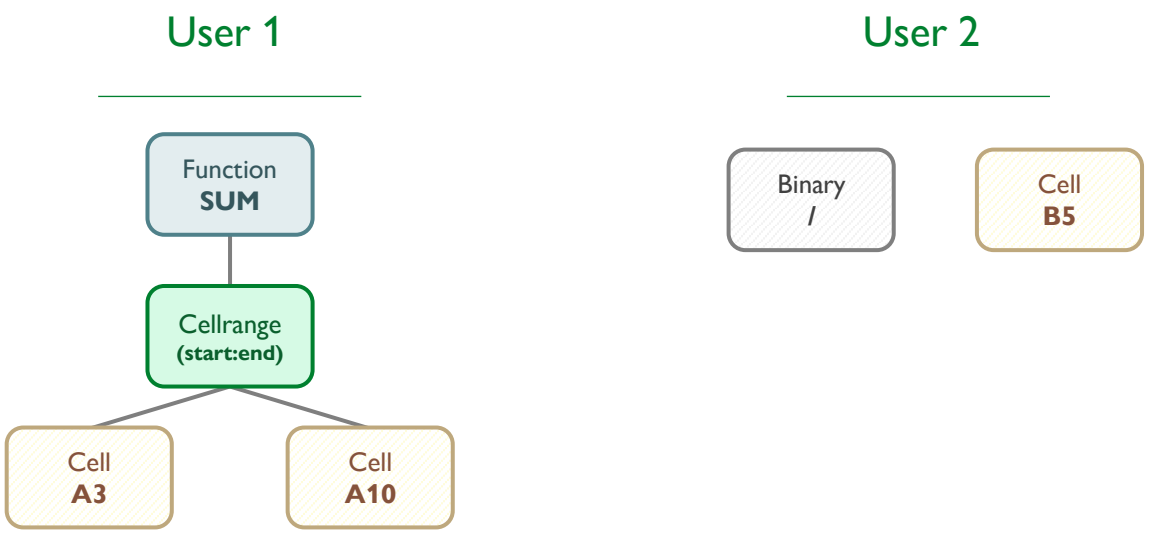
User 2



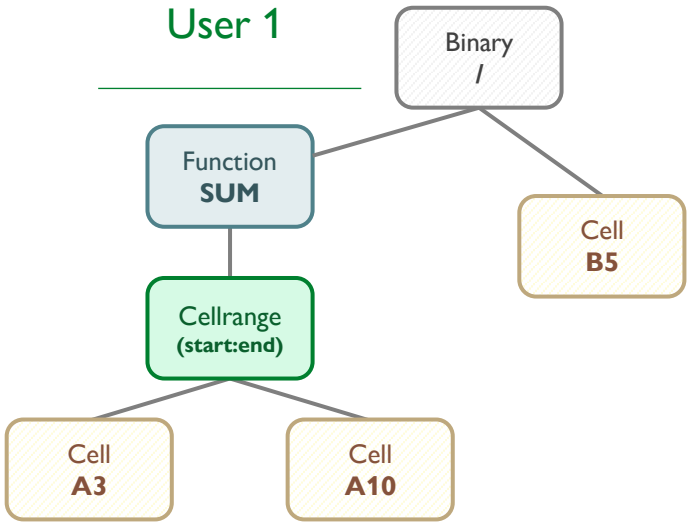
Apply User Changes



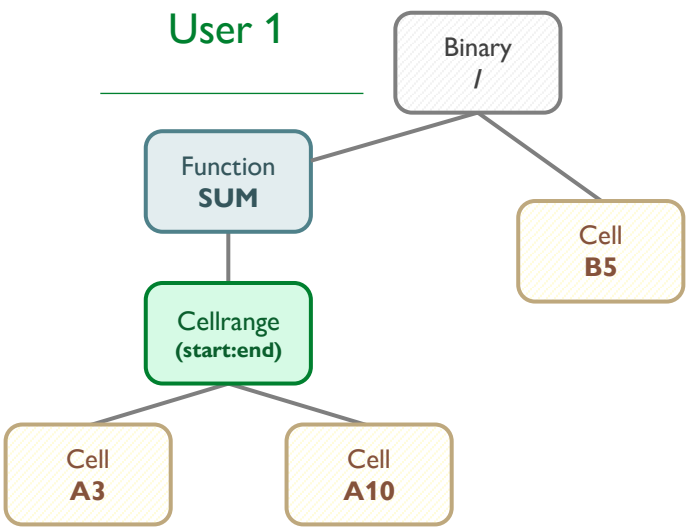
Apply User Changes

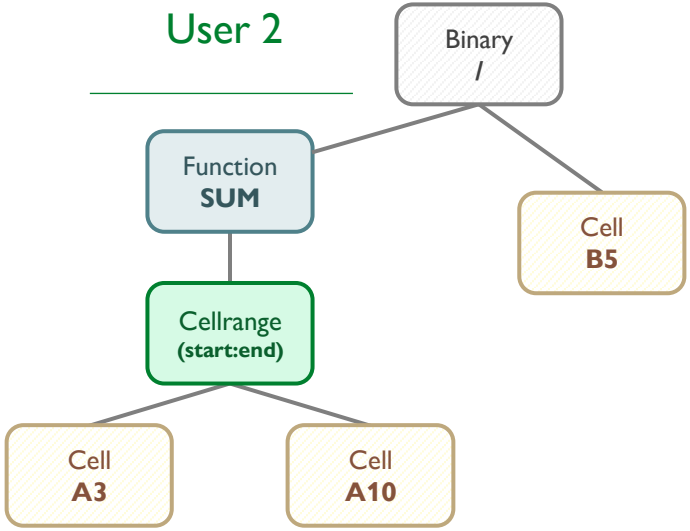


Apply User Changes



Both Users End Up With The Same AST!





Key Findings

What Were Our Key Findings?

Comparison

Approach 1: Rule-Based Method

- ✓ Effective for node-level decisions

  (Allows tailored merging strategies for specific scenarios)

- ✗ Struggles to cover all possible tree-level cases

Approach 2: History-Based Method

- ✓ Efficiently manages tree-level changes

  (Handles all scenarios comprehensively)

- ✗ Less precise for individual node-level conflicts

Final Implementation

Addition On Top

Approach 1:
Rule-Based Method

(Foundation)

Approach 2:
History-Based Method

Final Solution

Approach 3:
Hybrid Method

- Effective for node-level decisions
- Efficiently manages tree-level changes

Benefits of using both approaches

Inclusion

Sometimes it's a optimized solution to include changes of both users on one Node to get better merging results (e.g. CellRanges)

Syntax Preservation

To preserve the Syntax, we need strict Rule. Approach 2 alone wouldn't guarantee that the Syntax gets preserved.

Edge Cases

Some edge cases in merging cannot be solved with Approach 2 alone. We need specific rules to solve them.

Quality Of Results

What Was The Quality Of Our Results?

Quality Of Results

Baseline Comparison

Compared to the Baseline (Microsoft Excel & Google Sheets) we are able to successfully merge formulas after users worked offline.

Merge Quality

Our system ensures high-quality merging for simple formulas and basic edits, accurately reconciling most changes based on their edit history

Edge Cases

Sometimes the algorithm still encounters difficulties with complex formulas and large-scale or unusual edits that involve substantial changes, deletions, or complete rewrites.

Quality Of Results – Local-First Principles & CRDT

Technology	1. Fast	2. Multi-device	3. Offline	4. Collaboration	5. Longevity	6. Privacy	7. User control
Cellster	✓	✓	✓	✓	✓	✓	✓

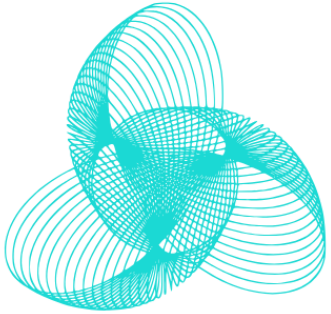
In distributed computing, a conflict-free replicated data type (CRDT) is **a data structure that is replicated across multiple computers in a network**, with the following features: The application can update any replica independently, concurrently and without coordinating with other replicas.

Thank You For Your Attention!

Q&A: Any Questions?

Feel Free To Come To Our Booth During The Apéro!

Cellster: Empowering Offline Collaboration In Spreadsheets – Demo At Our Booth!



Cellster

GitHub

Empowering Offline Collaboration in Spreadsheets

Break free from the chains of constant connectivity with the power of local-first software. Dive into a revolutionary way of collaboratively editing spreadsheet formulas without the need for real-time online presence. Experience seamless merges using CRDTs and witness the power of structured conflict resolution.