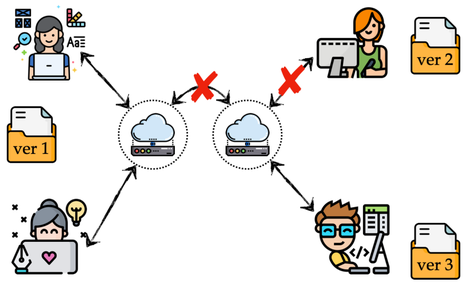
Certified MRDTs - Road to robust, decentralised apps

Vimala Soundarapandian — IIT Madras
Adharsh Kamath — NITK Surathkal
Kartik Nagar — IIT Madras
KC Sivaramakrishnan — IIT Madras

Local first software

Airtable · Google Docs · Notion · Overleaf



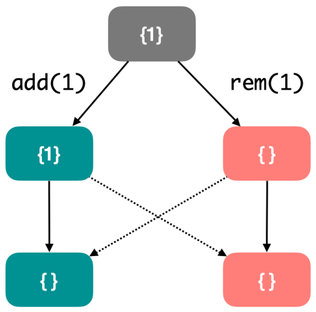
Mergeable Replicated Data Types (MRDTs)

- MRDTs - Inspired by Distributed Version Control Systems like Git
- 3-way merge uses the two versions to be merged and the lowest common ancestor (LCA) where the versions branched off
- Sequential data types + 3-way merge = replicated data type!

Motivating example

Observed Remove Set MRDT

OR-Set - add wins when there is concurrent add and remove of the same element



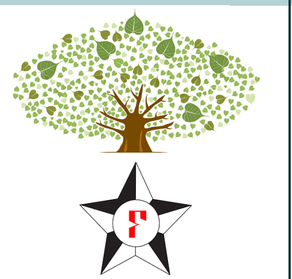

```
let merge ~lca ~v1 ~v2 =
  (lca ∩ v1 ∩ v2) (* unmodified elements *)
  ∪ (v1 - lca) (* added in v1 *)
  ∪ (v2 - lca) (* added in v2 *)
```

{} ∪ ({1} - {1}) ∪ ({} - {1})
= {} ∪ {} ∪ {}
= {} (expected {1})

- Convergence is not sufficient
- Intent is not preserved ☹

Our contribution (PEEPUL)

- PEEPUL - Certified MRDTs
- An F* library implementing and proving MRDTs
- Specification language is event-based
- Replication-aware simulation to connect specification with implementation



Verification approach

Specification

$$\mathcal{F}_{orset}(\text{rd}, \langle E, oper, rval, time, vis\rangle) = \{a \mid \exists e \in E.\ oper(e) = \text{add}(a) \wedge \neg(\exists f \in E.\ oper(f) = \text{remove}(a) \wedge e \xrightarrow{vis} f)\}$$

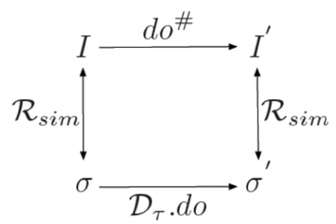
Simulation relation

*(connects **abstract state** with the **concrete state**)*

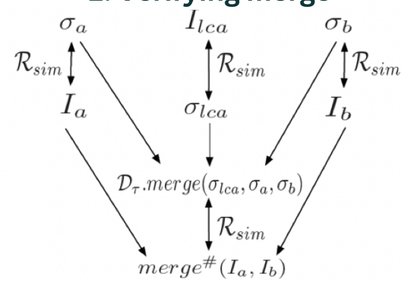
$$\mathcal{R}_{sim}(I, \sigma) \iff (\forall (a,t) \in \sigma \iff (\exists e \in I.E \wedge I.\ oper(e) = add(a) \wedge I.time(e) = t \wedge \neg(\exists f \in I.E \wedge I.\ oper(f) = remove(a) \wedge e \xrightarrow{vis} f)))$$

Proving data type implementations correct

1. Verifying operations



2. Verifying merge



3. Implementation satisfying the specification

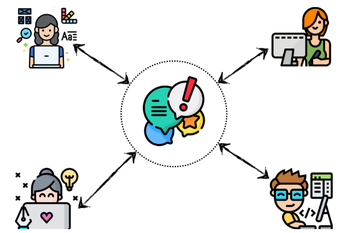
$\Phi_{spec}(\mathcal{R}_{sim})$	$\forall I, \sigma, e, op, a, t.\ \mathcal{R}_{sim}(I,\sigma) \wedge do^{\#}(I, e, op, a, t) = I' \wedge \mathcal{D}_\tau.do(op, \sigma, t) = (\sigma', a) \wedge \Psi_{ts}(I) \implies a = \mathcal{F}_\tau(o, I)$

4. Convergence

$(\mathcal{R}_{sim})$	$\forall I, \sigma_a, \sigma_b.\ \mathcal{R}_{sim}(I,\sigma_a) \wedge \mathcal{R}_{sim}(I,\sigma_b) \implies \sigma_a \sim \sigma_b$

Composing IRC style group chat

- IRC app state is constructed by instantiating a generic map with a mergeable log
- The proof of correctness of chat application directly follows from composition



Verification effort

MRDTs verified	#Lines code	#Lines proof	#Lemmas	Verif. time (s)
Increment-only counter	6	43	2	3.494
PN counter	8	43	2	23.211
Enable-wins flag	20	58	3	1074
		81	6	171
		89	7	104
LWW register	5	44	1	4.21
G-set	10	23	0	4.71
		28	1	2.462
		33	2	1.993
G-map	48	26	0	26.089
Mergeable log	39	95	2	36.562
OR-set (§2.1.1)	30	36	0	43.85
		41	1	21.656
		46	2	8.829
OR-set-space (§2.1.2)	59	108	7	1716
OR-set-spacetime	97	266	7	1854
Queue	32	1123	75	4753

Ongoing work

- Verifying MRDTs with reference sequential type and light-weight ordering constraint

References

[1] Gowtham Kaki, Swarn Priya, KC Sivaramakrishnan, and Suresh Jagannathan. 2019. Mergeable replicated data types. Proceedings of the ACM on Programming Languages 3, OOPSLA (Oct 2019), 1–29. https://doi.org/10.1145/3360580

[2] Sebastian Burckhardt, Alexey Gotsman, Hongseok Yang, and Marek Zawirski. 2014. Replicated Data Types: Specification, Verification, Optimality. In Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '14). ACM, New York, NY, USA, 271–284. https://doi.org/10.1145/2535838.2535848

[3] https://www.fstar-lang.org/