Lenses for Partially-Specified States

(accepted at ESOP 2026)

Kazutaka Matsuda (Tohoku University)
Minh Nguyen (University of Bristol)
Meng Wang (University of Bristol)

2026-03-09 @ PPL 2026

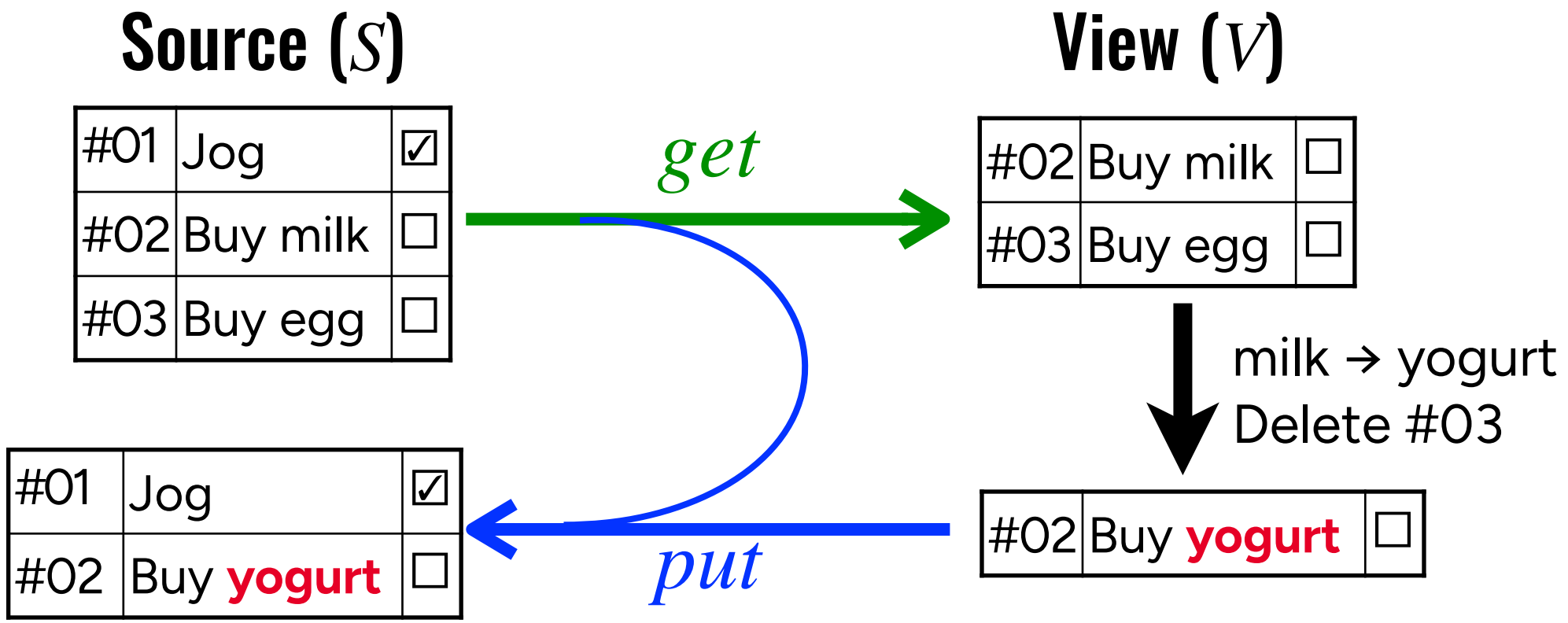
This work is about:

- Lens (bidirectional transformation) framework
  - ***view-to-view update propagation*** [Hu+ 04, Mu+ 04]
    - A change to one view triggers changes to other views
  - ***compositional guarantee*** of ***update preservation***
  - working on ***partially-ordered update intentions***
    - hybrid of states and operations
      - finer control on update propagation
    - inspired by CRDTs [Shapiro+11, Almeida+18]

Background: Lens

[Bancilhon&Spyratos 81, Hegner 90, Foster+ 05, 07, ...]

- A pair of $get : S \to V$ and $put : S \times V \to S$



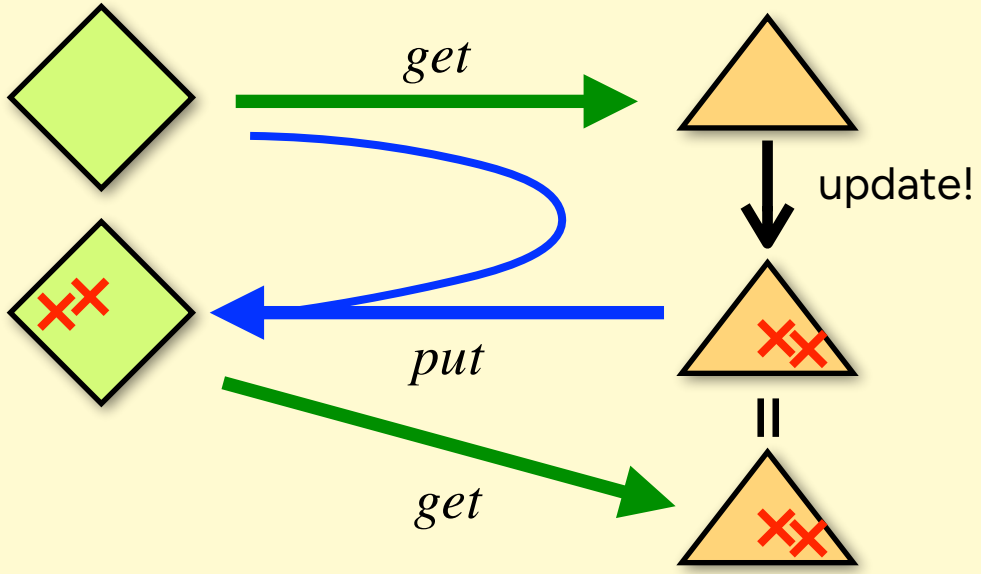
Well-Behavedness

[Bancilhon&Spyratos 81, Hegner 90, Foster+ 05, 07, ...]

Consistency (PutGet)

Today's focus

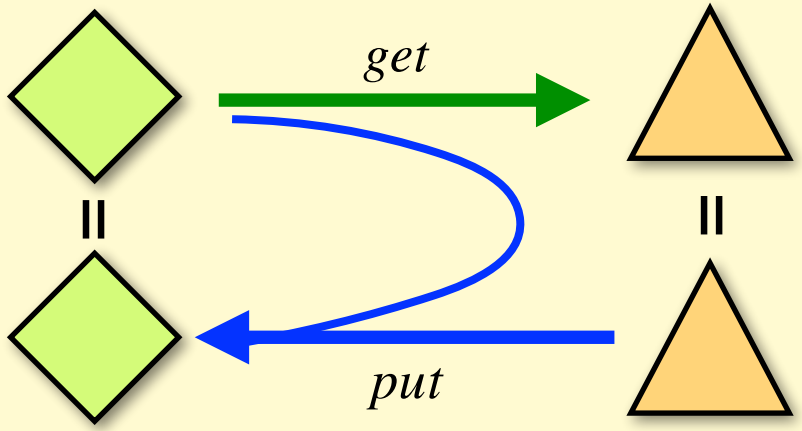
Put preserves user's update



$$put\ (s_0, v) = s \Longrightarrow get\ s = v$$

Acceptability (GetPut)

No update on the view,
no update on the source

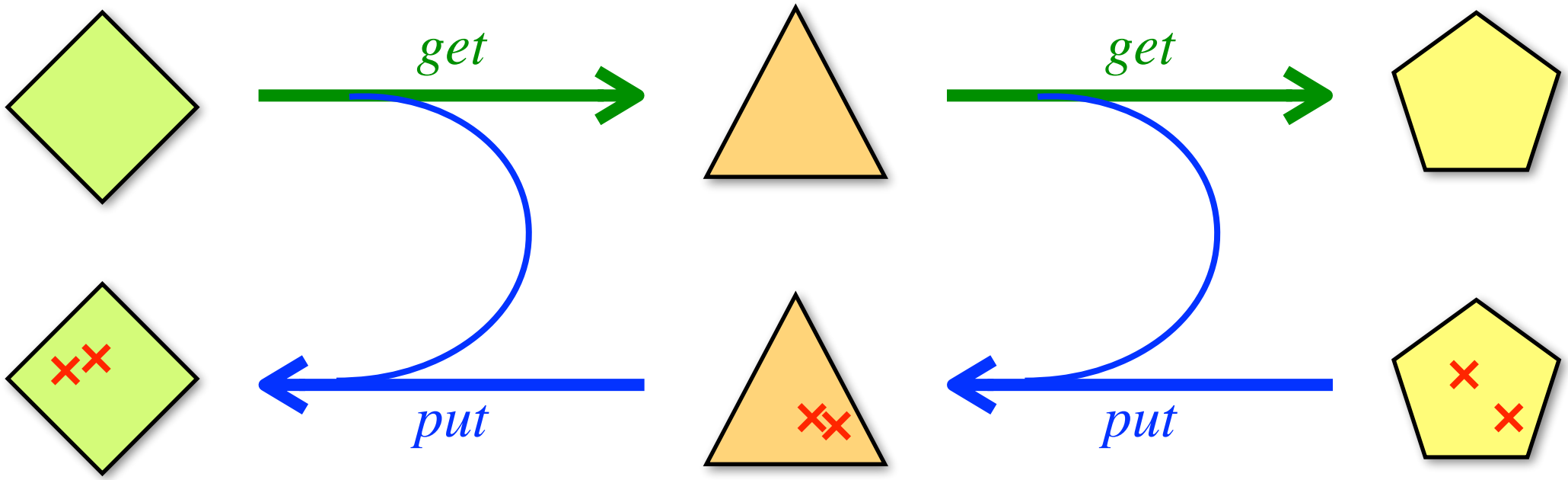


$$get\ s = v \Longrightarrow put\ (s, v) = s$$

Strength of Lens: Compositional Reasoning

- Well-behavedness is preserved by various combinators
  - especially, lens composition

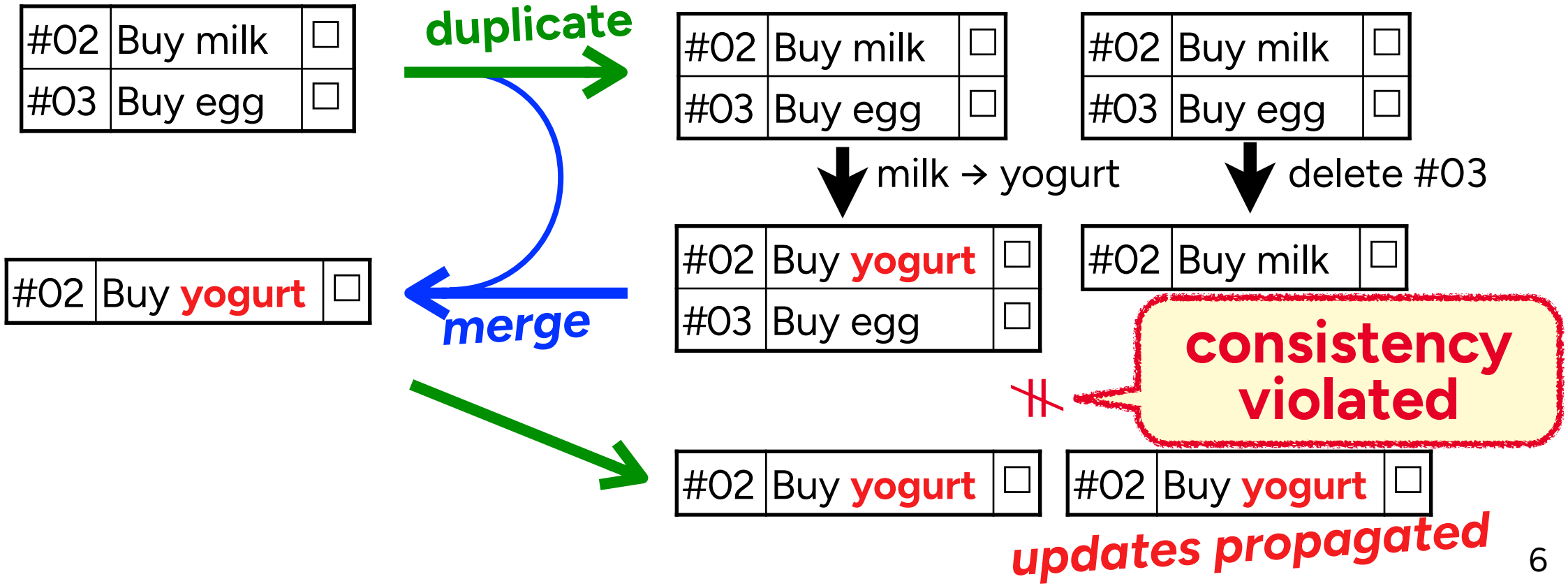
[Foster+ 05, 07]



Issue: View-to-View Update Propagation

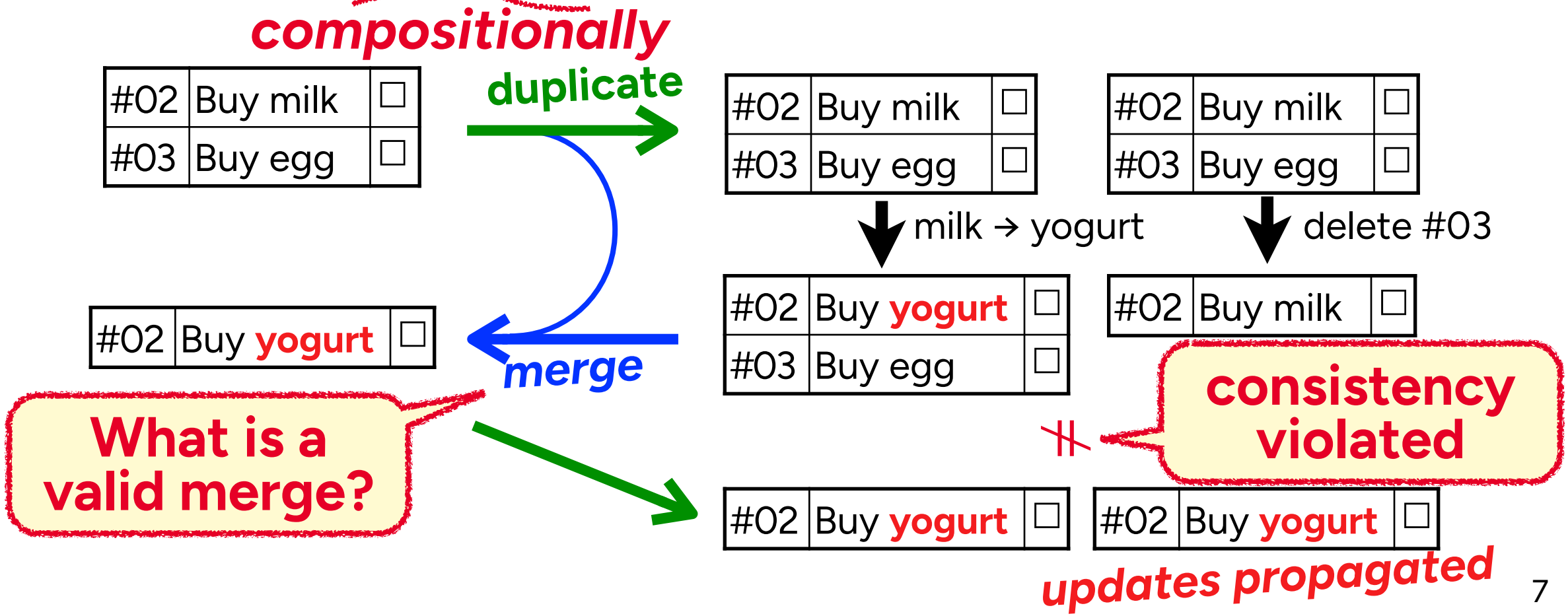
[Hu+ 04, Mu+ 04, ...]

- Duplication lens



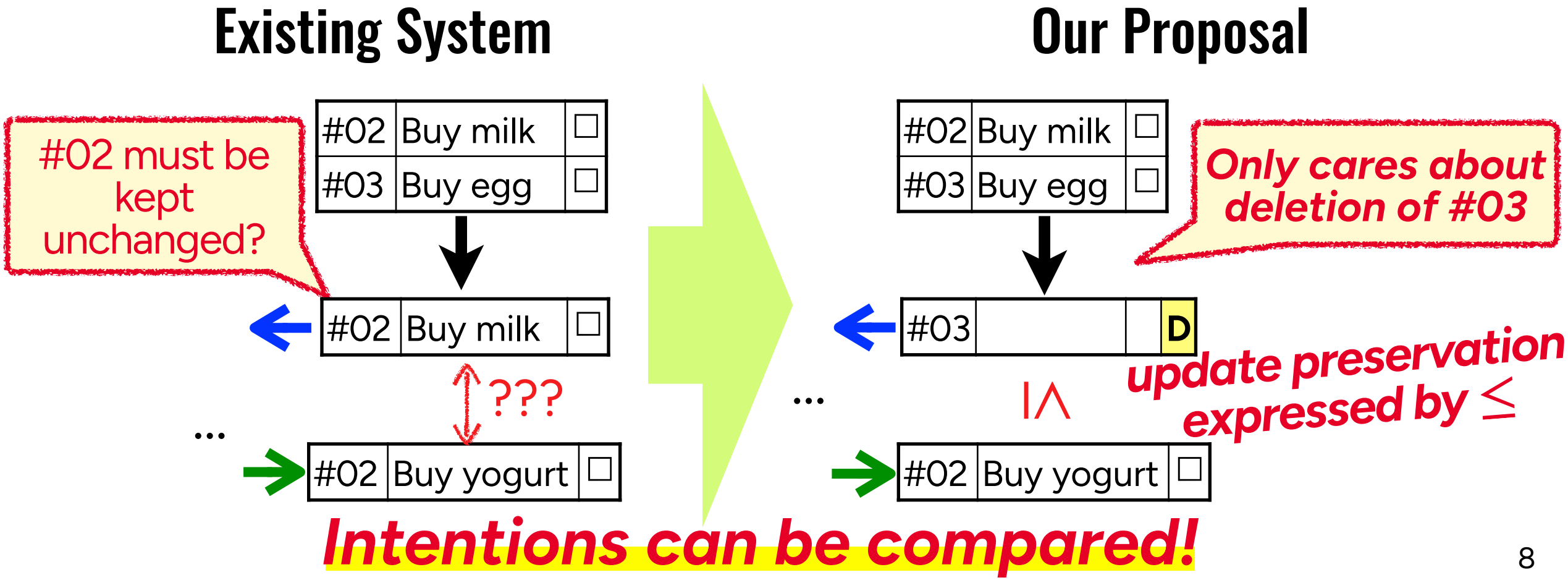
Challenge: Compositional Update Preservation

- *How to state and ensure update preservation?*



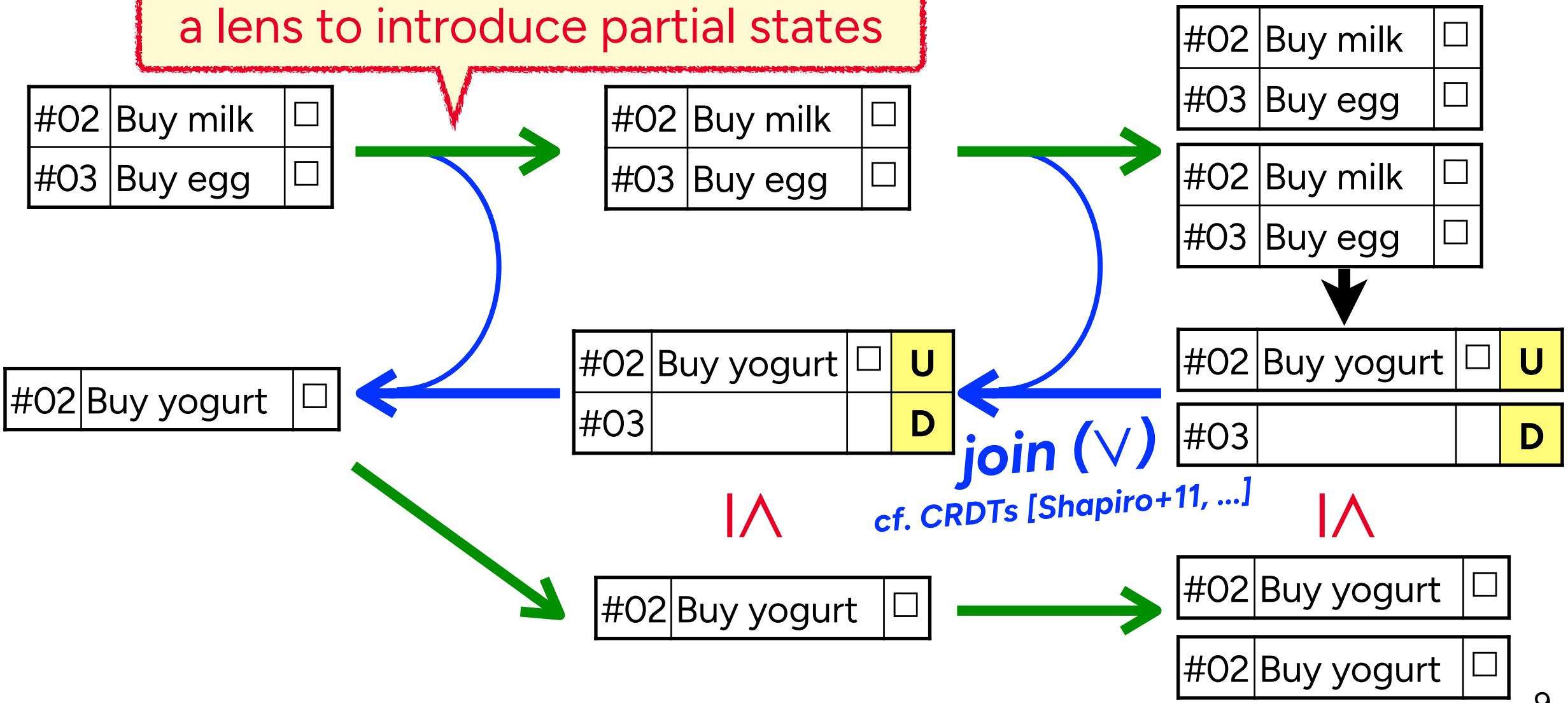
Our Approach: Partially-Ordered States

- Partially-ordered states representing ***update intentions***



Merge via Join (∨)

a lens to introduce partial states



Contributions

This talk!

- ***Partial-state lenses***: Lenses for partial intentions
  - ***view-to-view update propagation via shared source***
  - ***compositional laws***, guaranteeing
    - ***update preservation***
    - synchronization after a round-trip
- An involved example demonstrating utility
- Encoding of operations as partial intentions
  - cf. edit lenses [Hofmann+ 12] and delta lenses [Diskin+ 11]

Outline

- Source and view domains
- Lawless partial-state lenses
- Our well-behavedness
  - ps-consistency (ps-PutGet)
  - ps-acceptability (ps-GetPut)
  - ps-stability (ps-PutGetPut)

Source and View Domains

- A poset $(S, \leq)$ with a reflexive subset $\leq_I$ of $\leq$

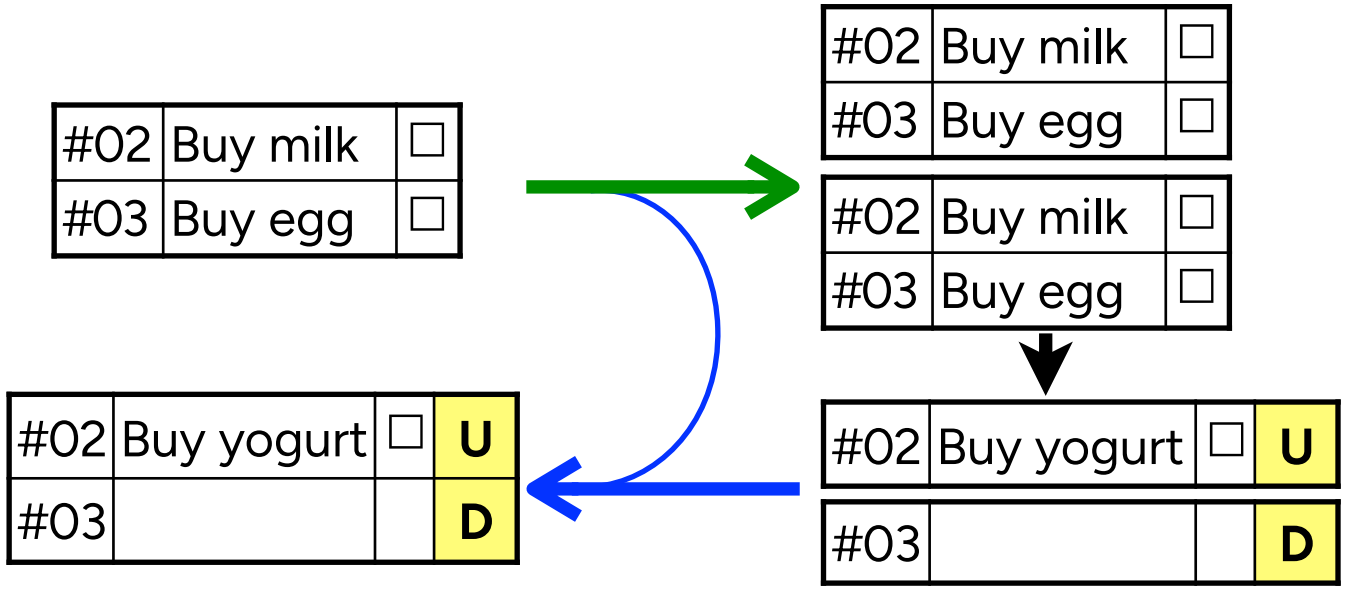
$\leq_I$: $\leq$ restricted to "no updates" (details omitted in this talk)

  - no built-in update reflection
  - no built-in distinction of "proper" states

Lawless Partial-State Lens

- A pair of $get : S \to V$ and $put : S \times V \to S$
  - no different from the original lens [Foster+ 05, 07]
  - … except S and V are posets (with $\leq_I$)

Example:
duplication lens



Outline

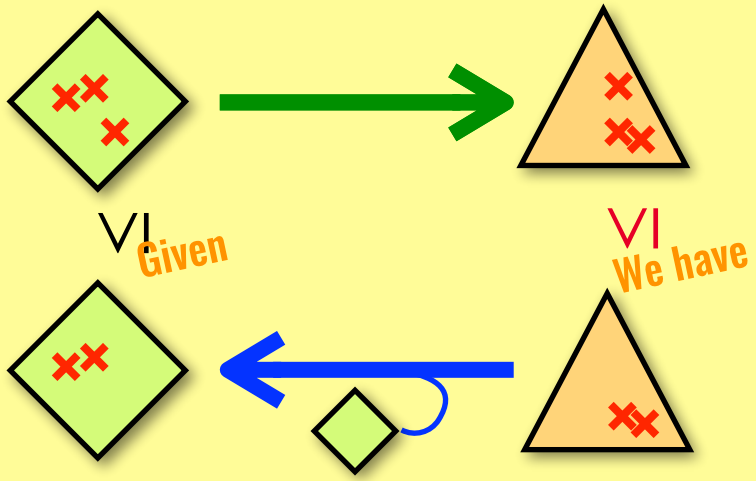
- Source and view domains
- Lawless partial-state lenses
- Our well-behavedness
  - ps-consistency (ps-PutGet)
  - ps-acceptability (ps-GetPut)
  - ps-stability (ps-PutGetPut)

Our Laws

This talk!

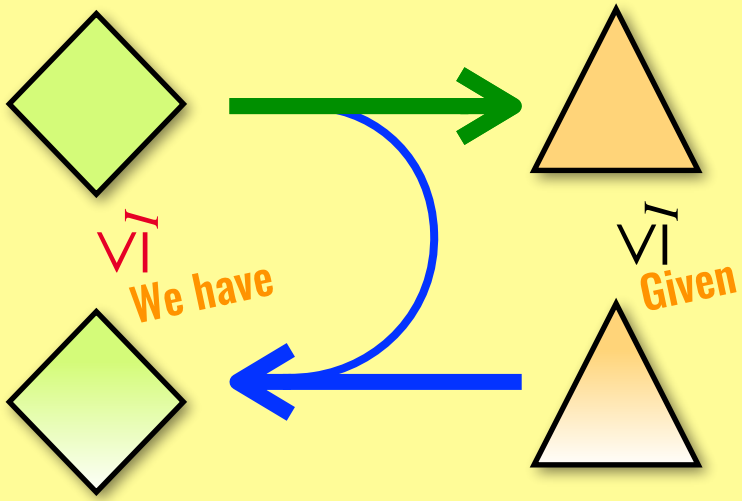
PS-Consistency

Put-then-get preserves user's update



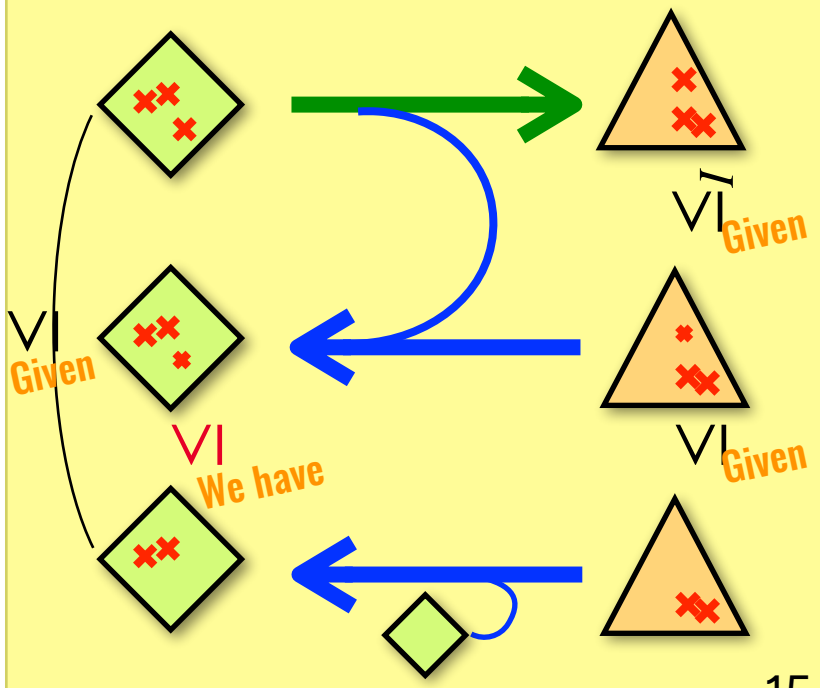
PS-Acceptability

No update on the view, no update on the source



PS-Stability

Put gives a “definite” result
(put-gen-put cannot refine it)

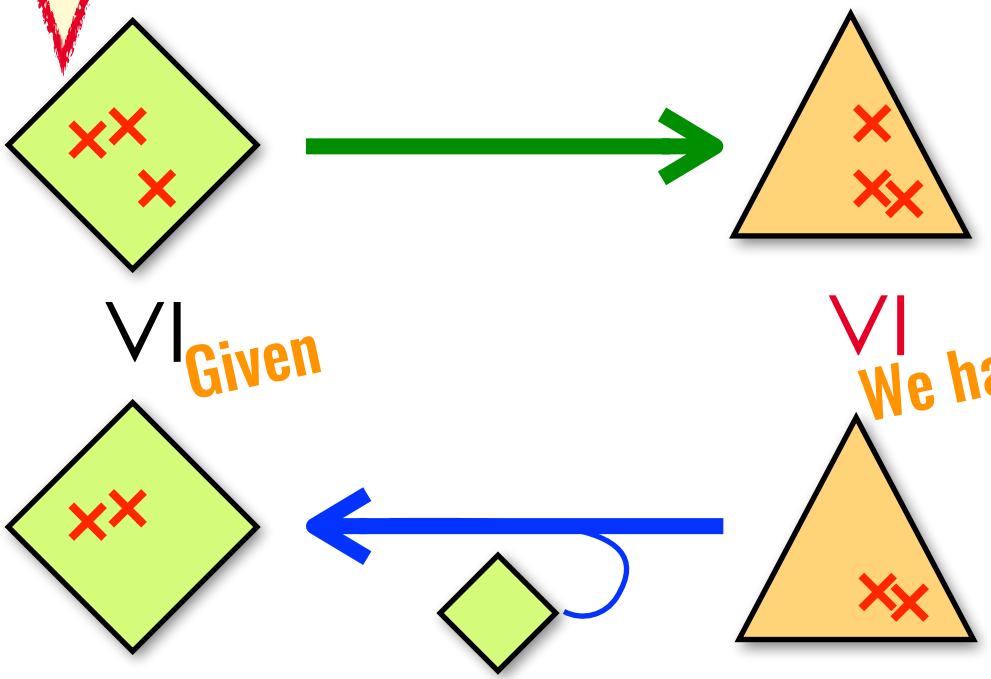


PS-Consistency

$$\frac{put\ (s_0, v) \leq s}{v \leq get\ s}$$

- Put-then-get preserves user's update

merge may introduce more updates

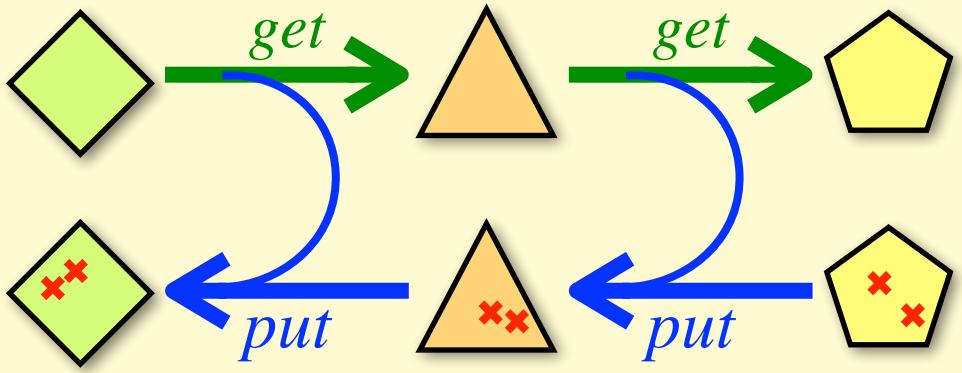


update preservation

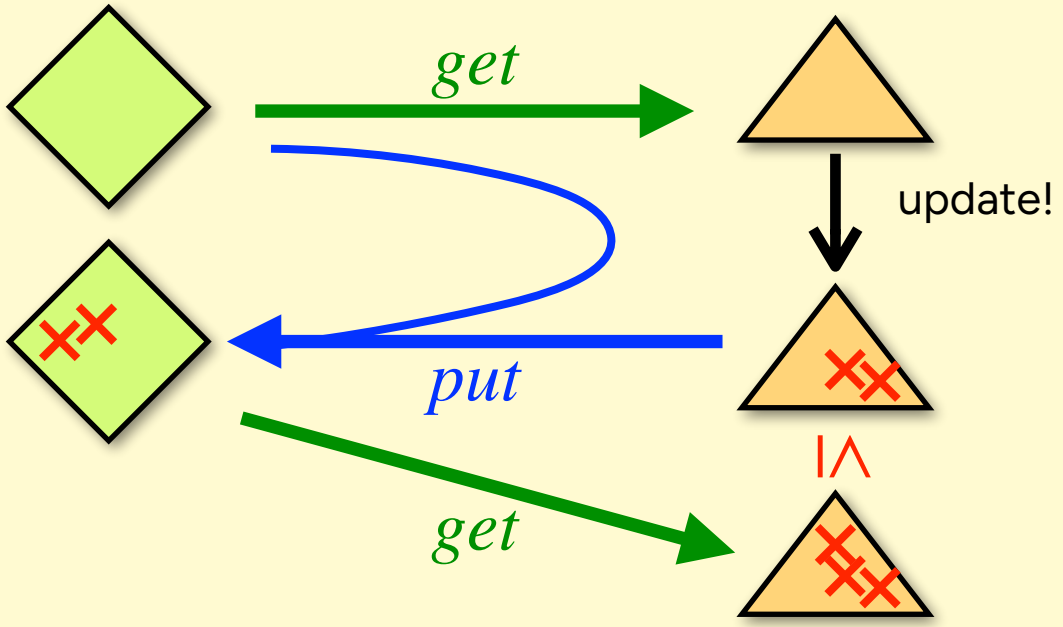
$(\leq) = (=)$ implies the original consistency (PutGet)

Theorems (See more in our paper...)

Our well-behavedness is
closed under lens composition



Our well-behavedness implies
update preservation



Related Work

- PutGetPut [Mu+ 04], WPutGet [Hidaka+ 10]
  - no compositional, no guarantee of update preservation
- $\preceq$-well-behavedness [M&W 18]
  - similar laws to ours, except for maximality conditions
  - limited forms of partial states and lenses
    - no view-to-view update propagation
- [Hegner 04]'s order-based laws
  - stronger than the ***very*** well-behavedness [Foster+ 05, 07]

no partial states in get

Conclusion

- Lens framework, called ***partial-state lenses***
  - view-to-view update propagation [Hu+ 04, Mu+ 04]
  - ***compositional guarantee*** of ***update preservation***
  - working on ***partially-ordered update intentions***
    - hybrid of states and operations
      - finer control on update propagation
    - inspired by CRDTs [Shapiro+11, Almeida+18]

Future Directions: more domains/lenses, how to input intentions