Development History Granularity Transformations

Kıvanç Muşlu[1][2] Luke Swart[3] Yuriy Brun[4] Michael D. Ernst[2]

[1] Microsoft, Tools for Software Engineers

[2] University of Washington, Computer Science & Engineering

[3] HaxGeo, Civic Software Development

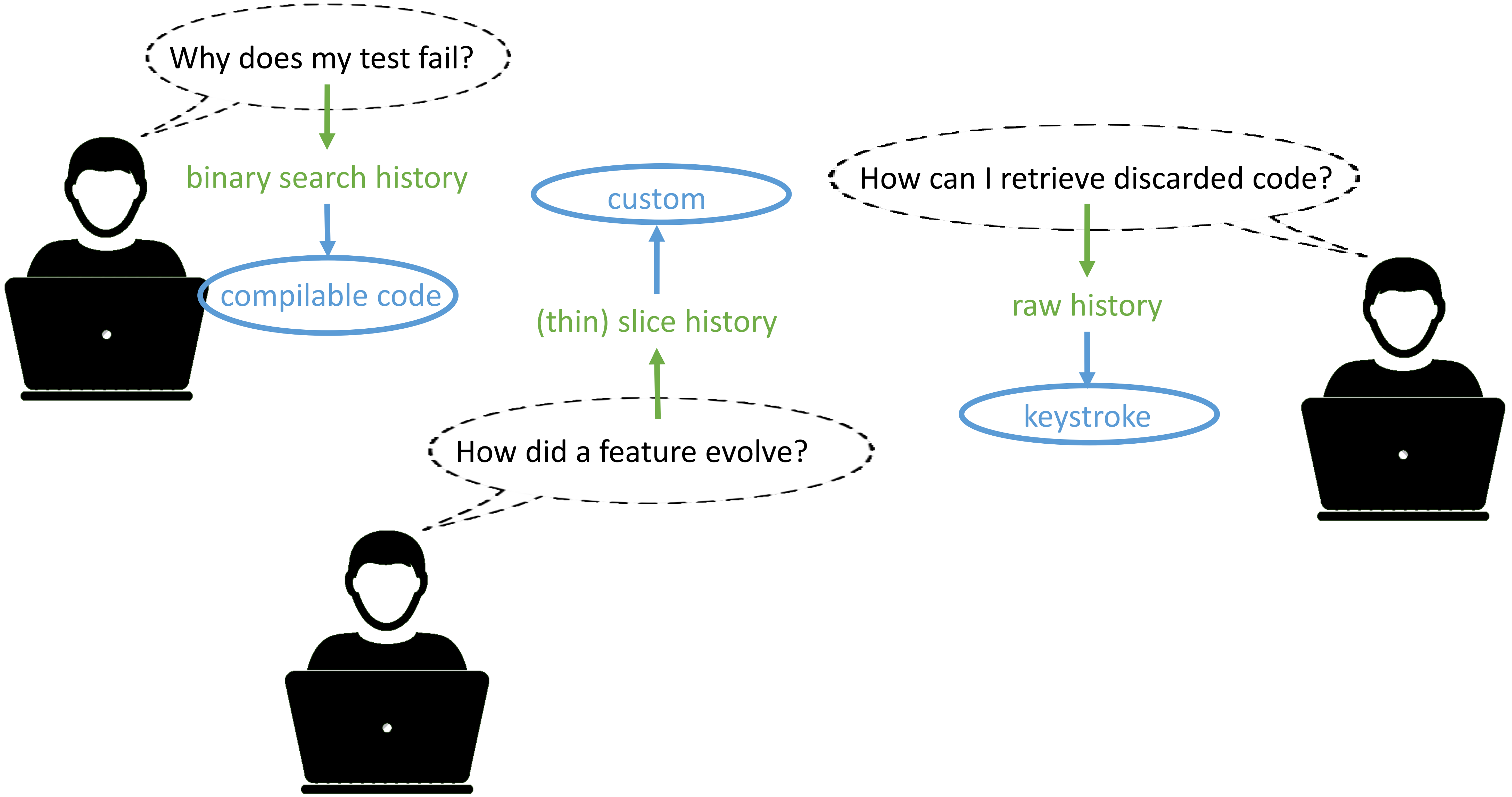
[4] University of Massachusetts Amherst, Information and Computer Science

Development histories simplify tasks

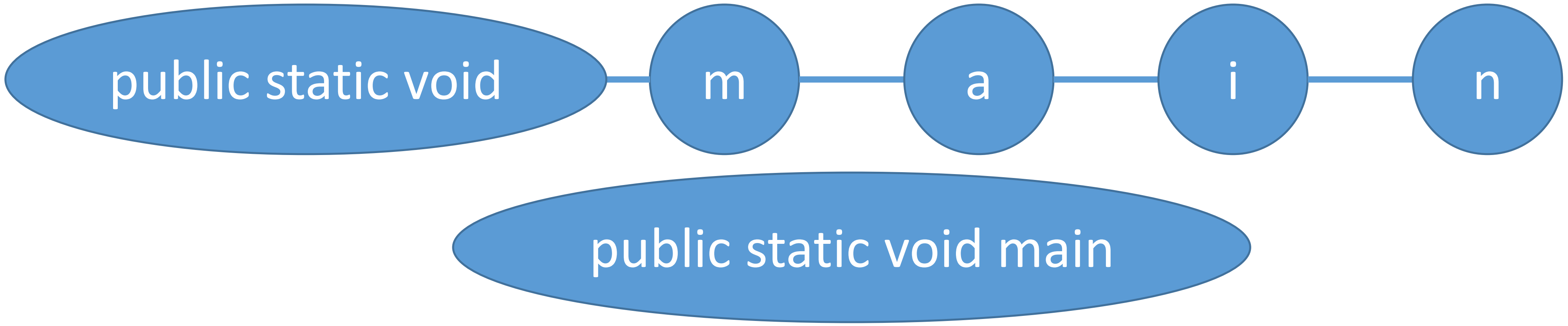
Development histories are used to:

- localize bugs
- rollback mistakes
- understanding software evolution
- predicting failures
- …

Different tasks require different granularities



Problem: development histories are inflexible



- automatically-managed histories
  [YoonM11, Mahoney12, NegaraCDJ14]
  - Fine-grained: extracting relevant information requires post processing
- manually-managed histories
  - Incomplete: might miss information
  - Course-grained: information might be intermingled with irrelevant one

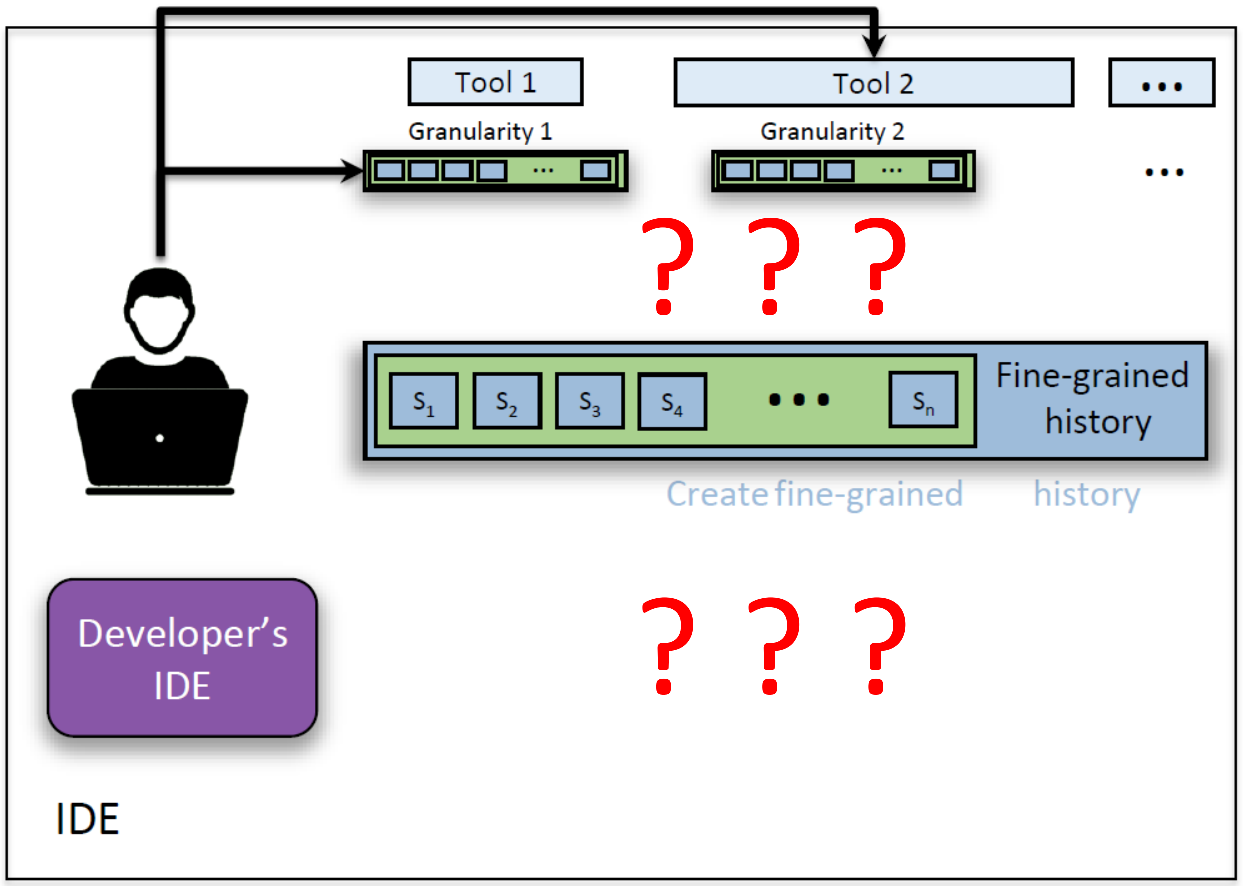
Solution: multi-grained development histories

Our contribution:

make recording granularity transparent

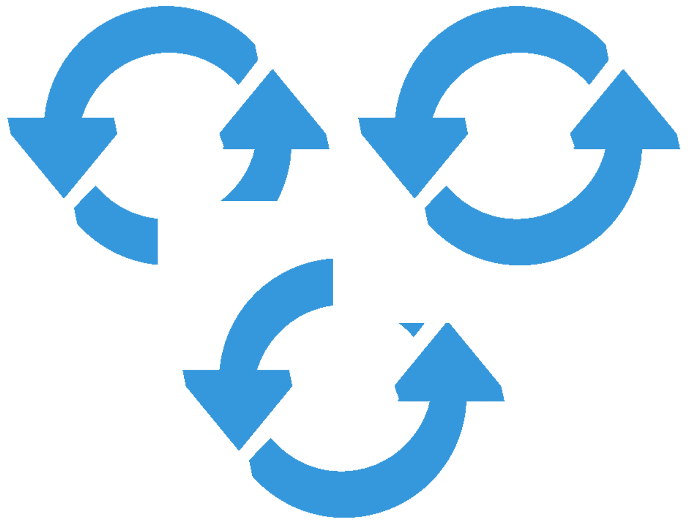
- record a complete & fine-grained history
- automatically transform this history into more optimal granularities for the task at hand

Solution: multi-grained development histories

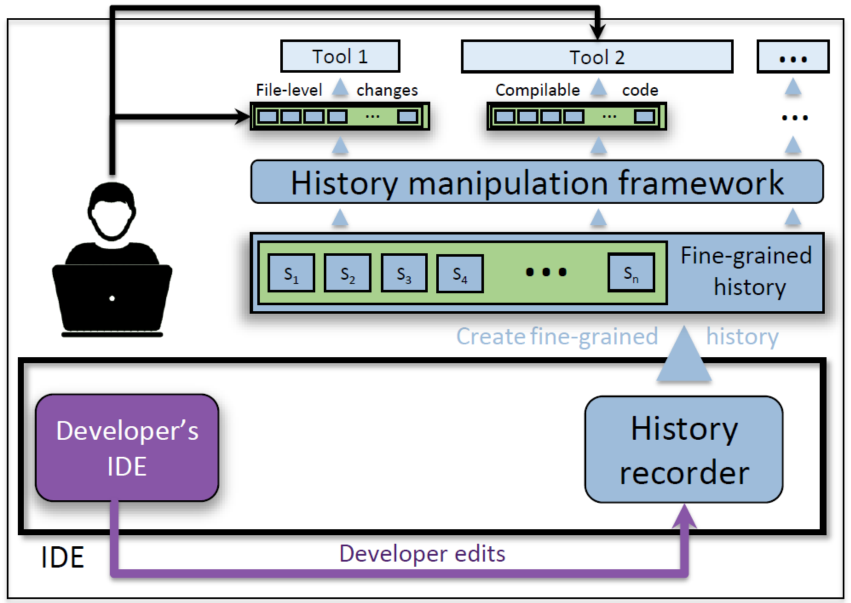


Outline

Transformations



Design



Transformations

granularity transformations

(group changes that satisfy … and reorder history such that …)

↑

transformation operations

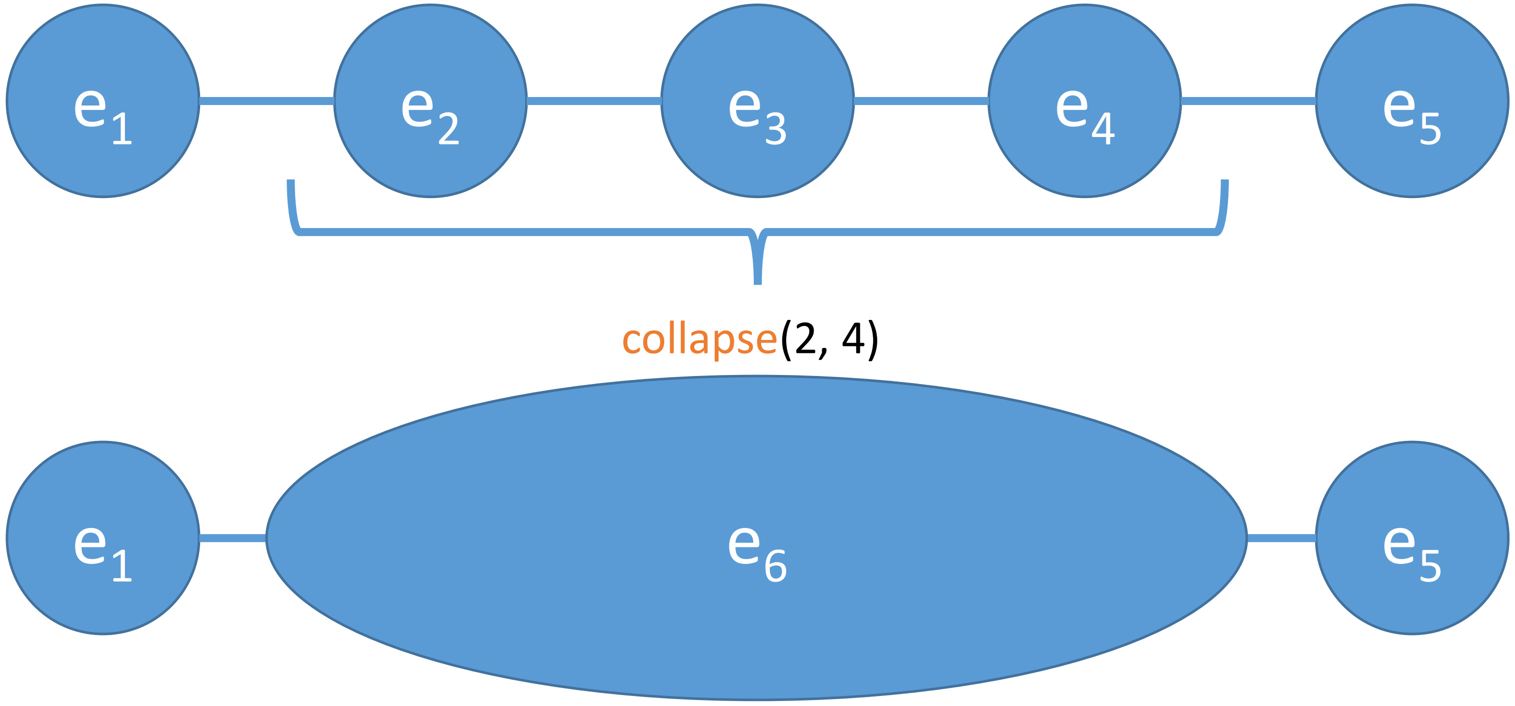
(intermediate operations)

↑

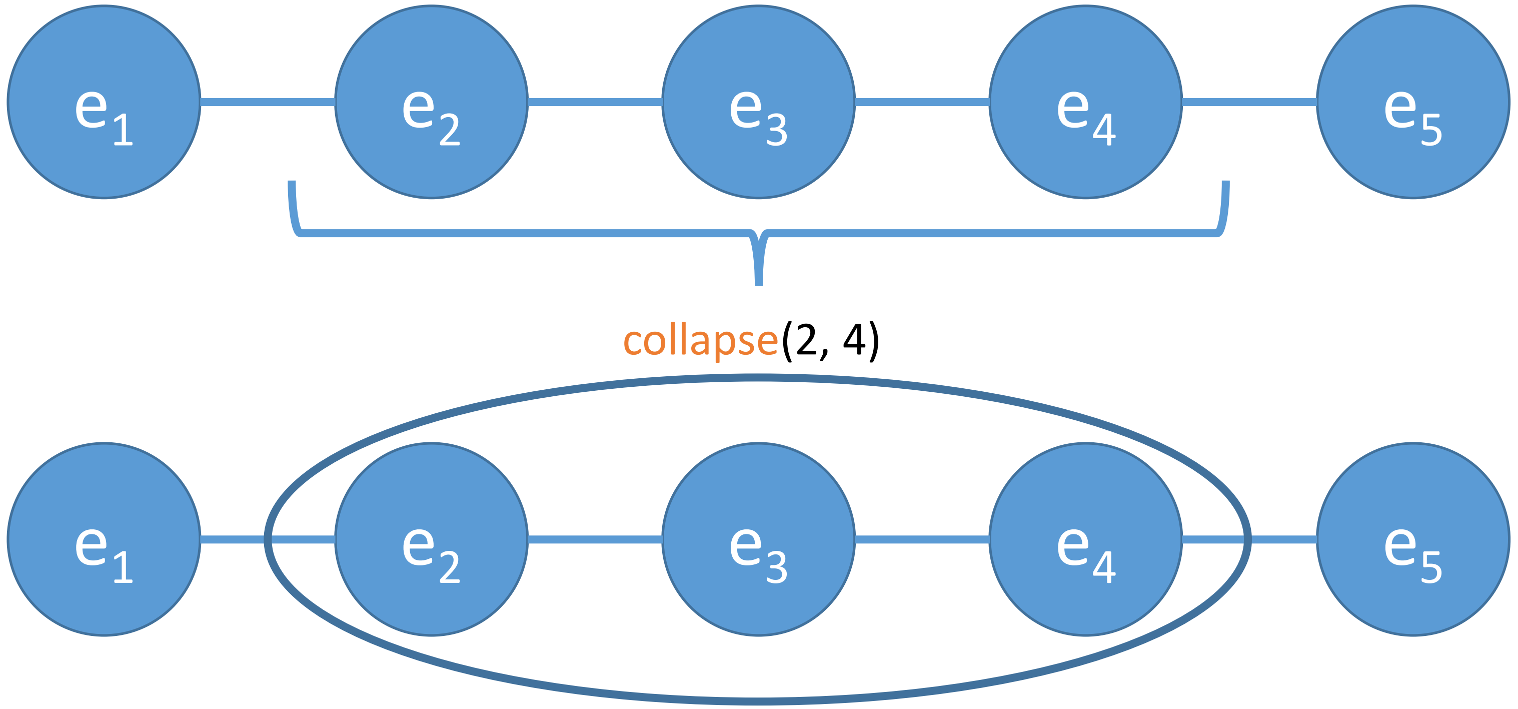
transformation primitives

expand, collapse, group

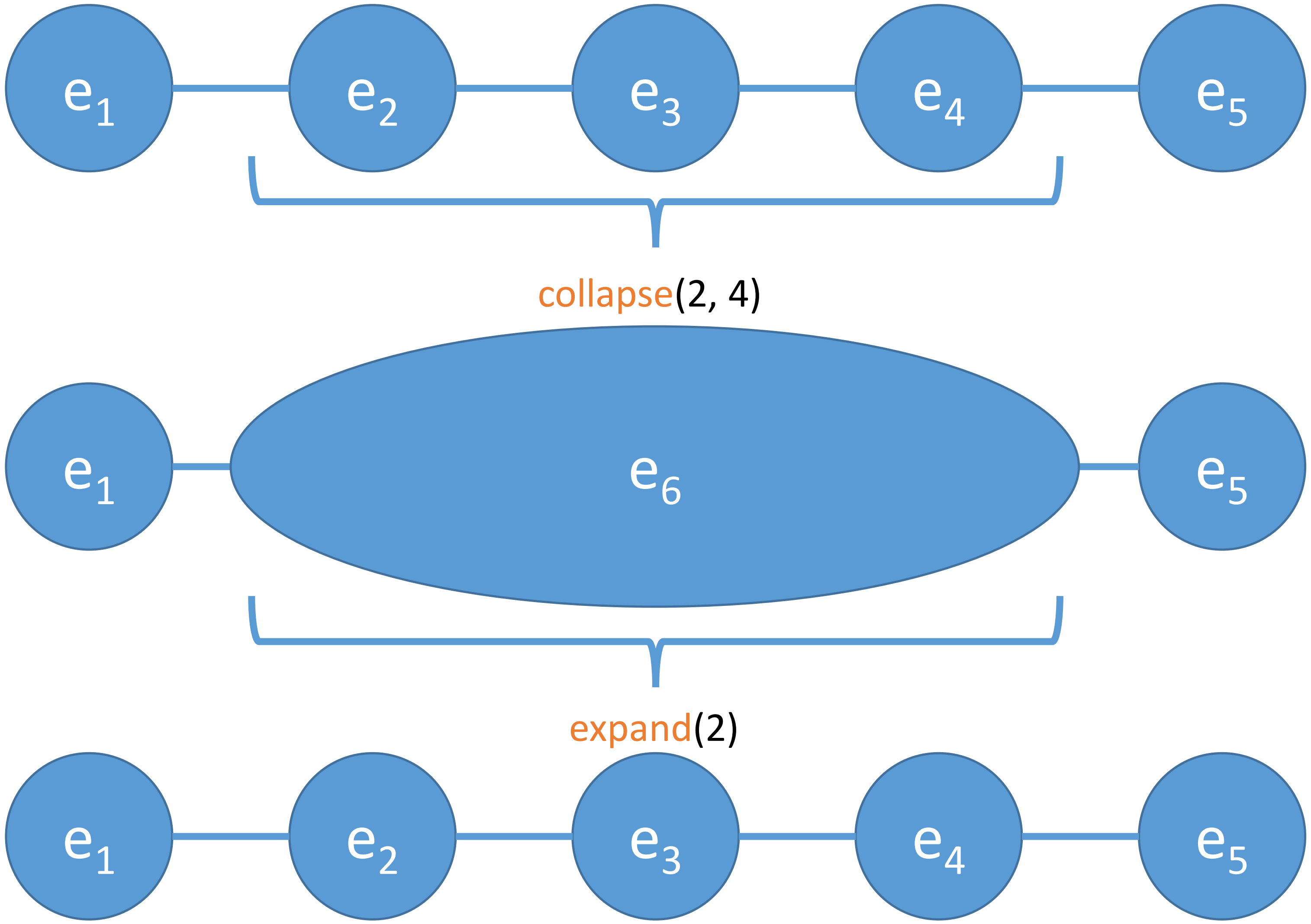
Primitives: expand, collapse, and move



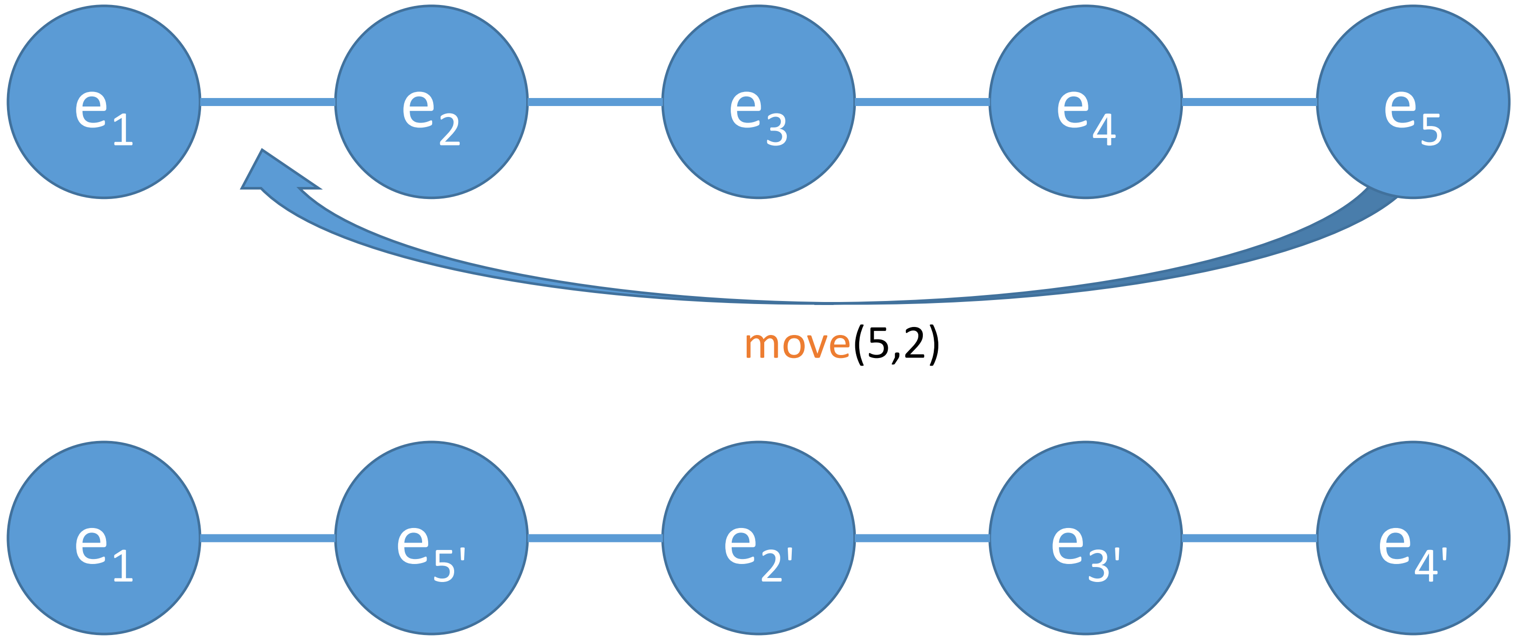
Primitives: expand, collapse, and move



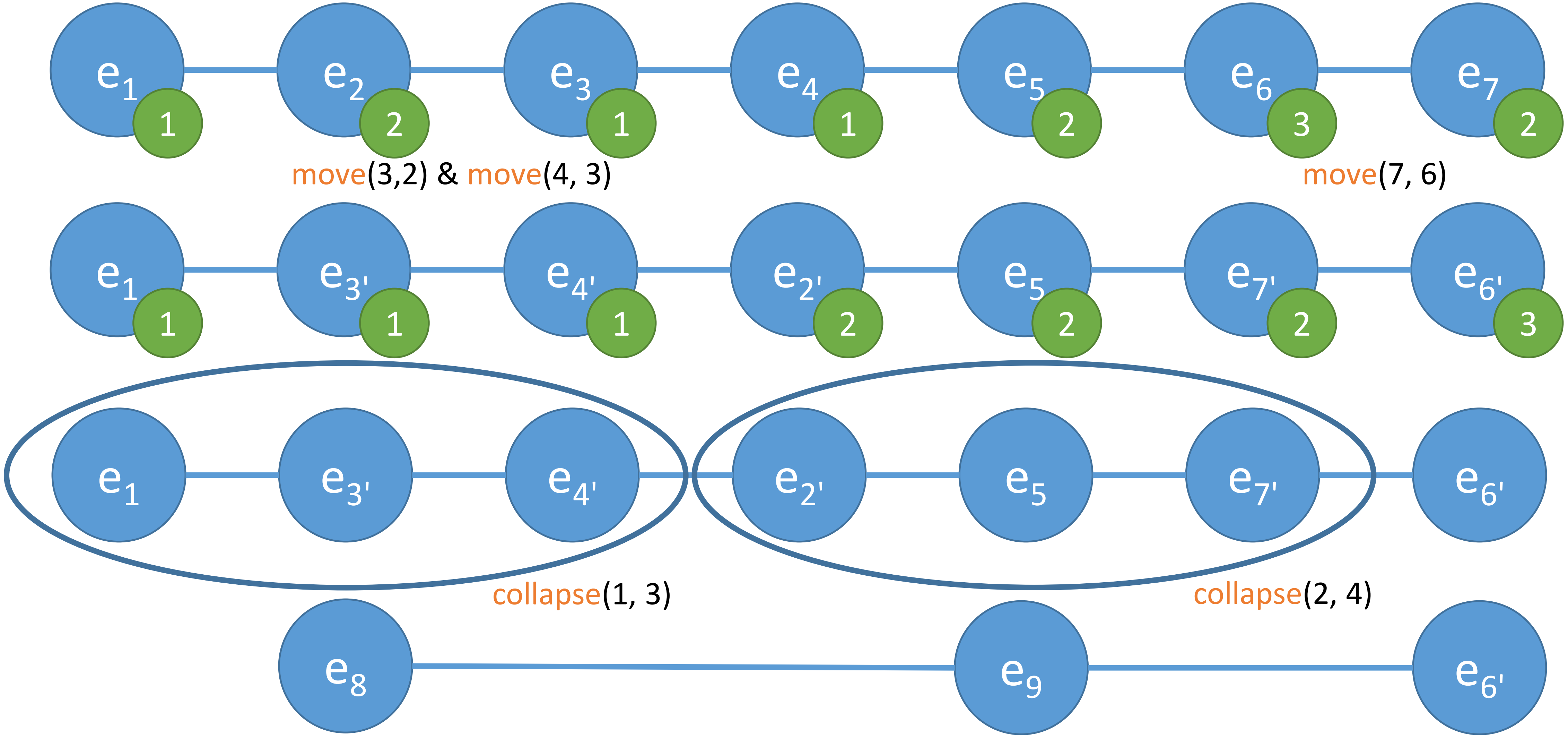
Primitives: expand, collapse, and move



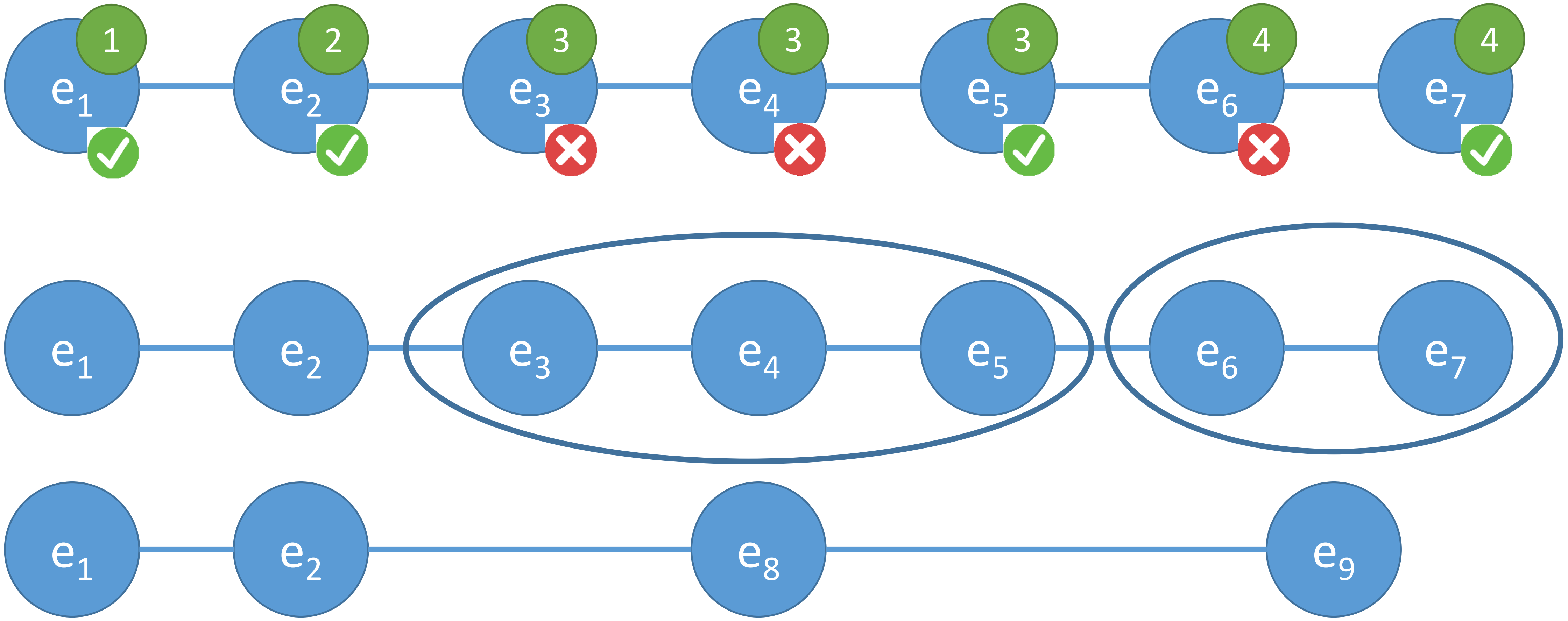
Primitives: expand, collapse, and move



Operation: group (move + collapse)



Transformation: GroupCompilable (group)



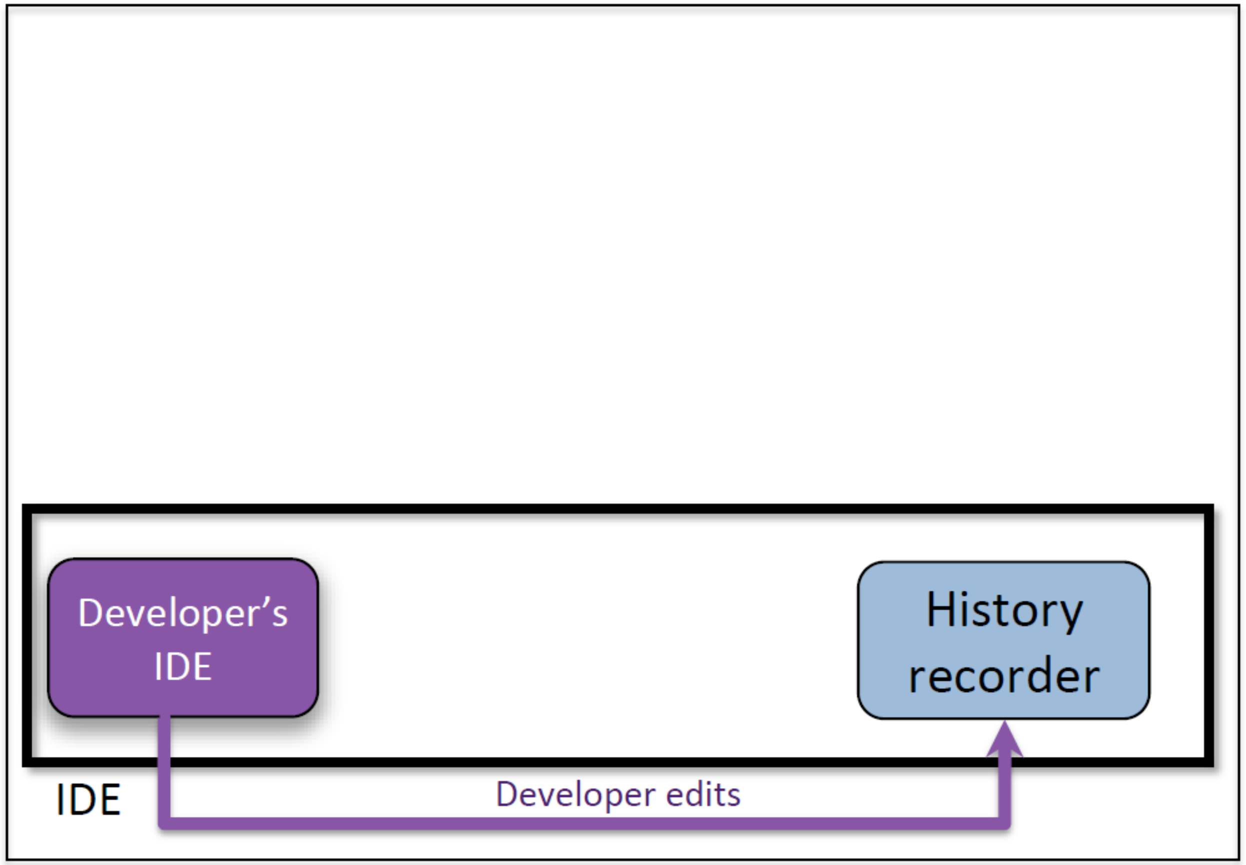
All transformations

- GroupCompilable: group(collapse)

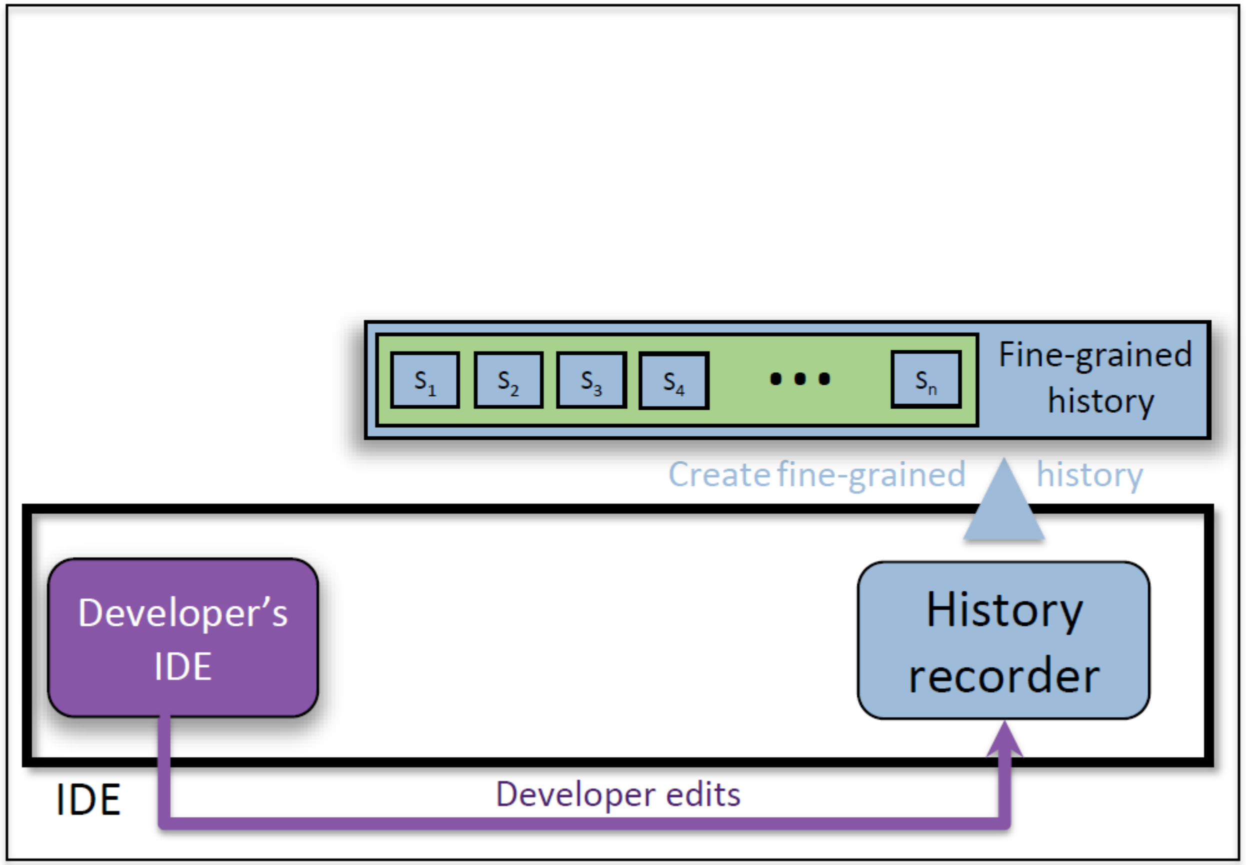
- GroupFiles: group(collapse + move)
  - for each modified file, creates a group containing all edits on this file
  - useful for manual inspection (e.g., VCS diff)

- GroupCollocated: expand + group(collapse + move)
  - creates a group for each contiguous edit
  - useful for separating tangled changes

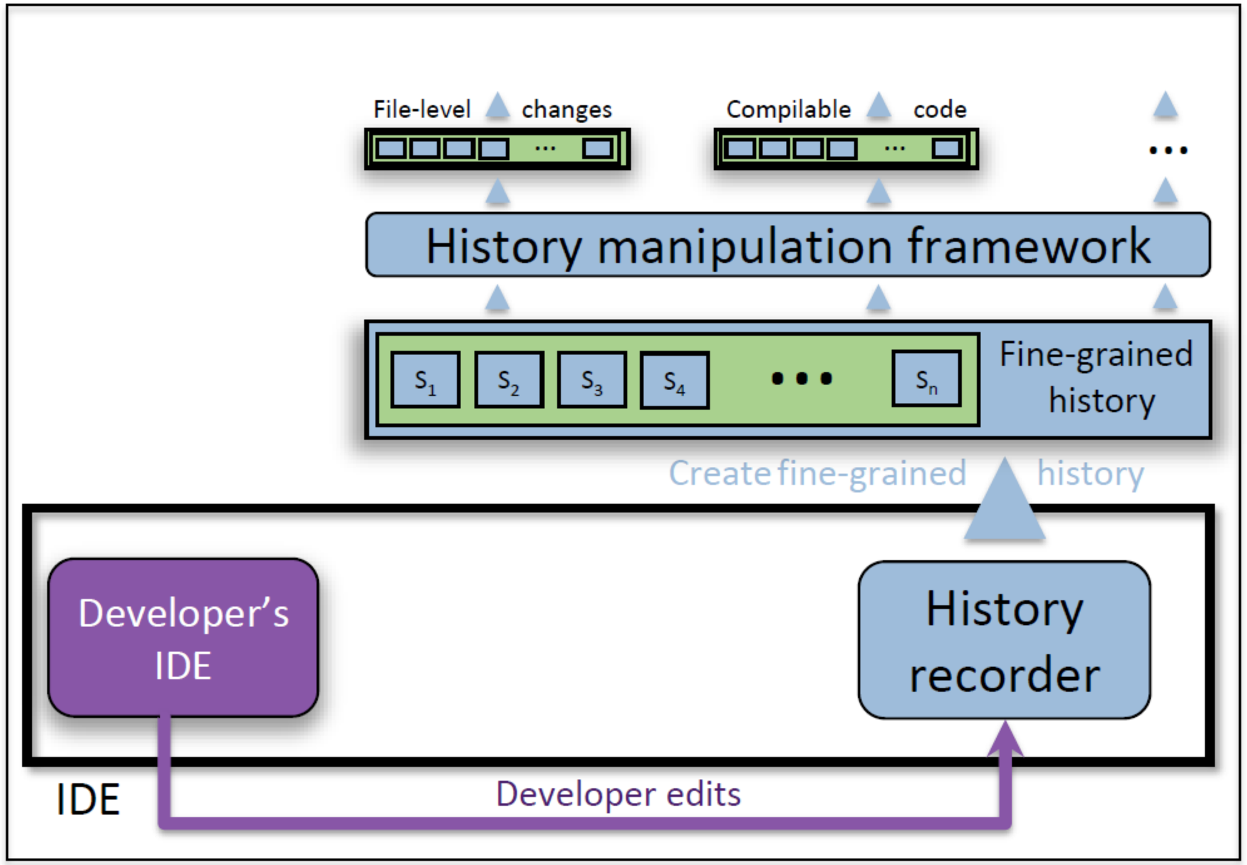
Codebase Manipulation:
a design for multi-grained histories



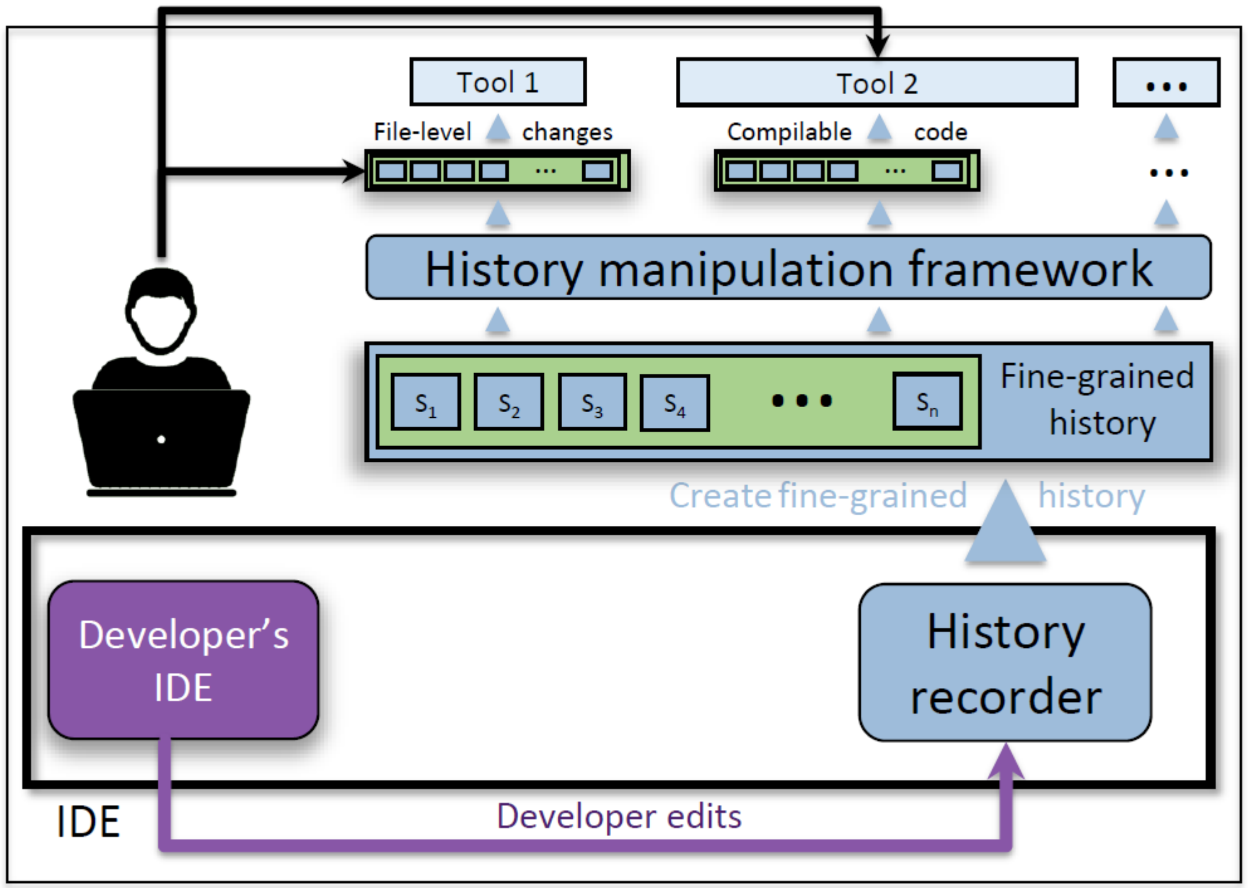
Codebase Manipulation:
a design for multi-grained histories



Codebase Manipulation: a design for multi-grained histories



Codebase Manipulation: a design for multi-grained histories



Contributions

- identify inflexibility problem of the current development histories
- propose multi-grained histories
  - Builds on three primitives: collapse, expand, move
  - History is automatically recorded
  - Developer uses the most optimal granularity for the current task
- Codebase Manipulation: one design for multi-grained histories