

Live & Local Schema Change

Challenge Problems

Jonathan Edwards, Tomas Petricek, Tijs van der Storm



CHARLES
UNIVERSITY



Centrum Wiskunde & Informatica



university of
groningen

Live & Local Schema Change

Challenge Problems

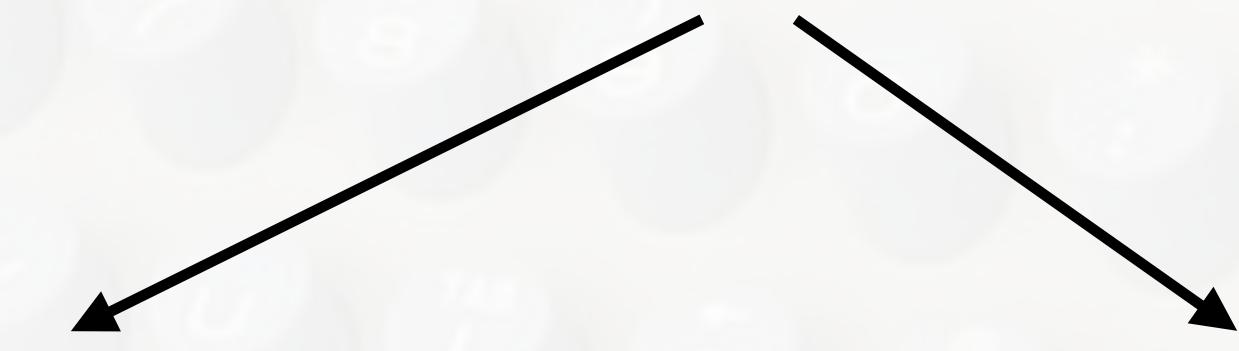
Jonathan Edwards, Tomas Petricek, Tijs van der Storm



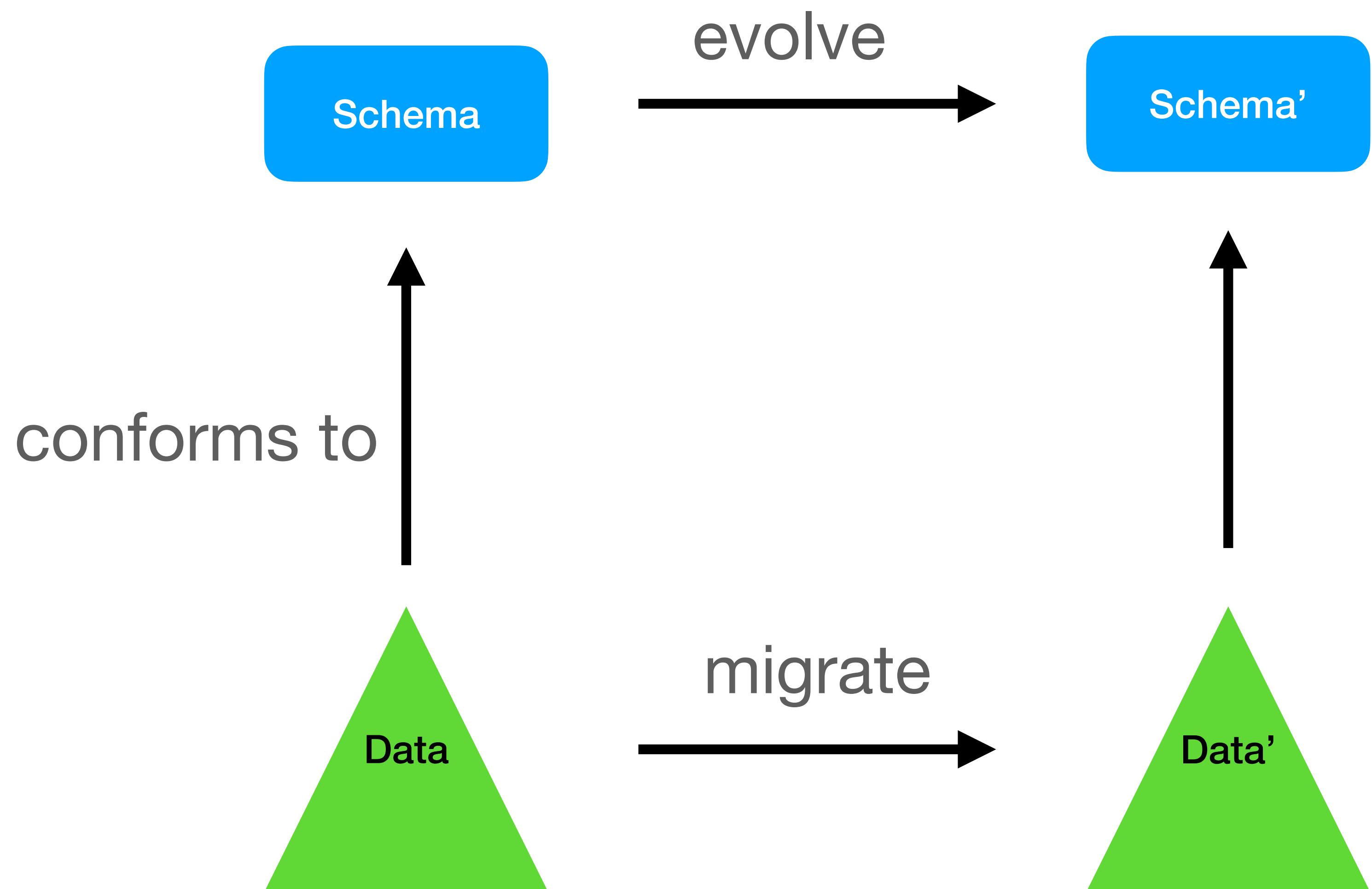
CHARLES
UNIVERSITY

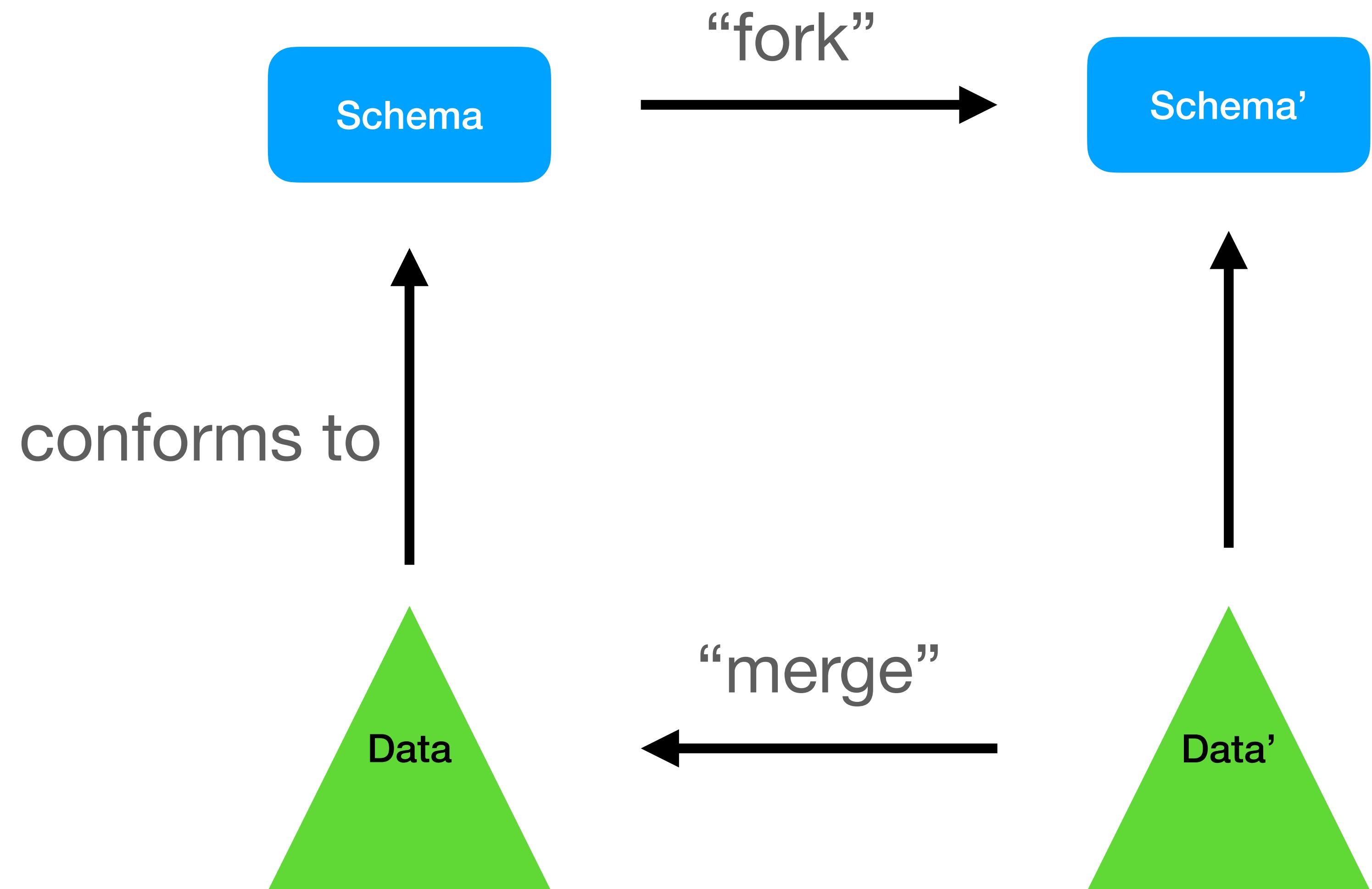


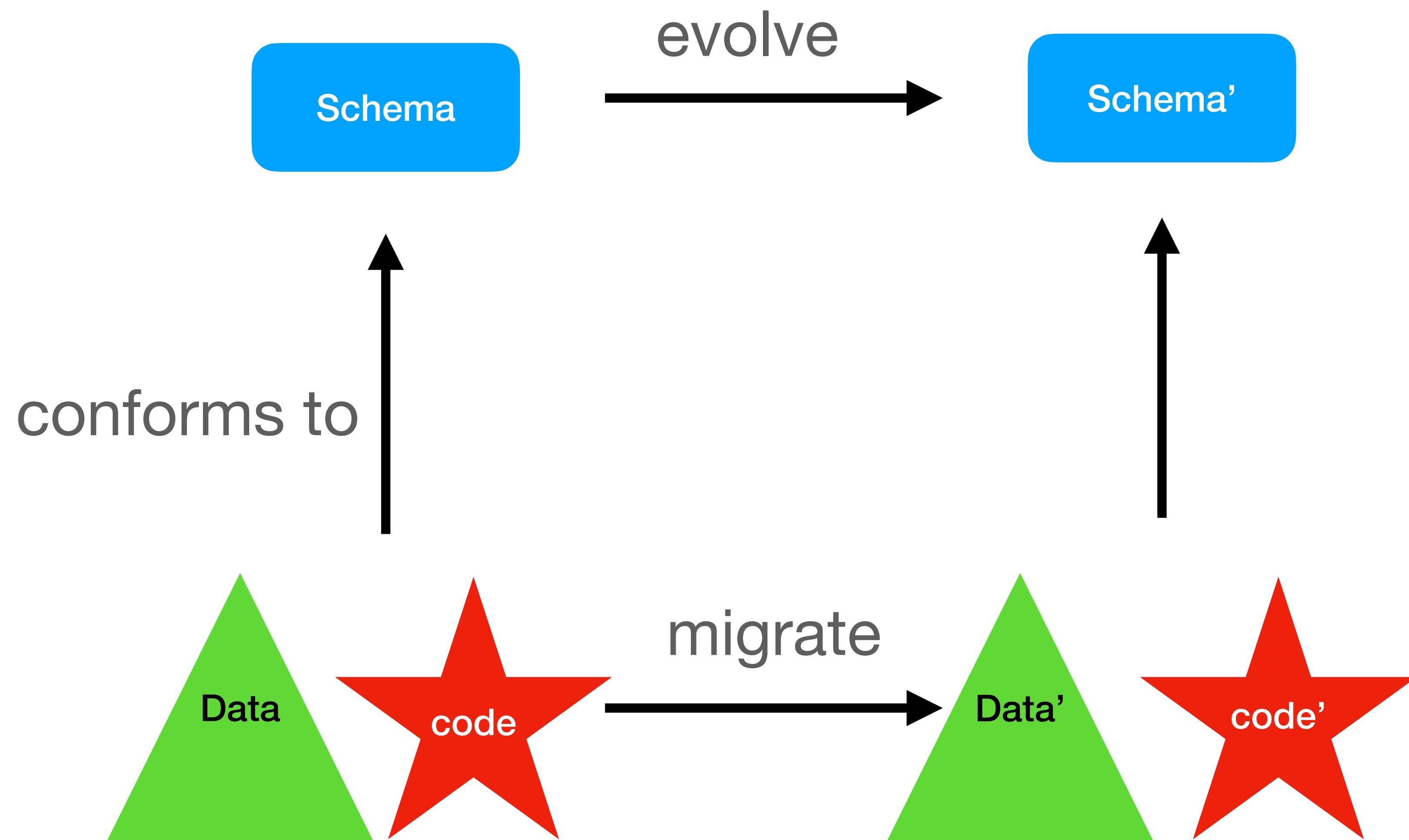
Centrum Wiskunde & Informatica



university of
groningen







Challenge problems

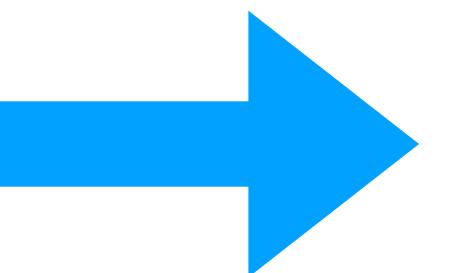
same same but from different application domains

- Databases: extract entity
- Spreadsheets: divergence control
- Structural documents: merging edits
 - + in the presence of code
- Live modeling: run-time state migration
- Live GUI programming: Elm

Extract entity

Before

Item	Quantity	Ship Date	Customer name	Customer address
Anvil	1	2/3/23	Wile E Coyote	123 Desert Station
Dynamite	2		Daffy Duck	White Rock Lake
Bird Seed	1		Wile E Coyote	123 Desert Station

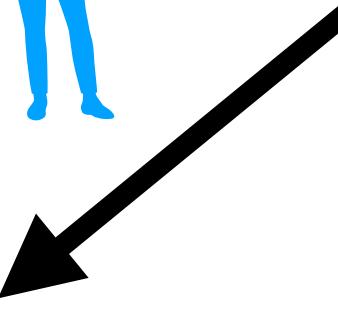


After

Item	Quantity	Ship Date	Customer name
Anvil	1	2/3/23	Wile E Coyote
Dynamite	2		Daffy Duck
Bird Seed	1		Wile E Coyote

Name	Address
Wile E Coyote	123 Desert Station
Daffy Duck	White Rock Lake

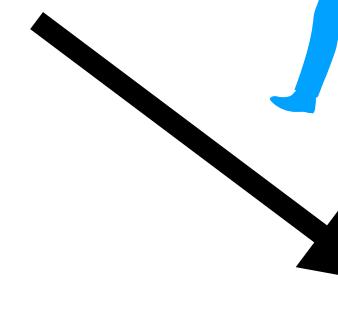
Merging structural document edits



PROGRAMMING CONFERENCE 2023

Invited speakers

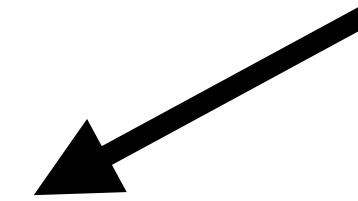
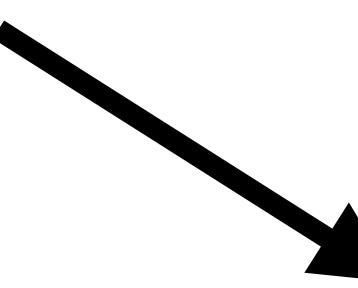
- Adele Goldberg, adele@xerox.com
- Margaret Hamilton, hamilton@mit.com
- Betty Jean Jennings, betty@rand.com



PROGRAMMING CONFERENCE 2023

Invited speakers

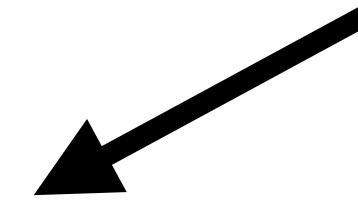
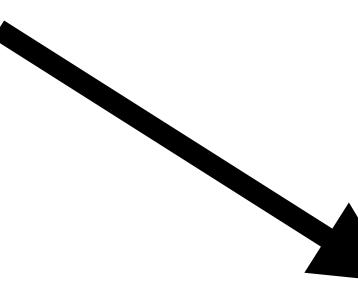
- Ada Lovelace, lovelace@rsoc.ac.uk
- Adele Goldberg, adele@xerox.com
- Betty Jean Jennings, betty@rand.com
- Margaret Hamilton, hamilton@mit.com



PROGRAMMING CONFERENCE 2023

Invited speakers

Name	Email	Organizer
Adele Goldberg	adele@xerox.com	TP
Margaret Hamilton	hamilton@mit.com	JE
Betty Jean Jennings	betty@rand.com	JE



PROGRAMMING CONFERENCE 2023

Invited speakers

Name	Email	Organizer
Ada Lovelace	lovelace@rsoc.ac.uk	
Adele Goldberg	adele@xerox.com	TP
Betty Jean Jennings	betty@rand.com	JE
Margaret Hamilton	hamilton@mit.com	JE

Merging structural document edits with code

PROGRAMMING CONFERENCE 2023

Invited speakers

- Adele Goldberg, adele@xerox.com
- Margaret Hamilton, hamilton@mit.com
- Betty Jean Jennings, betty@rand.com

Conference budget

Travel cost per speaker:

\$1200

Number of speakers:

=COUNT(/ul[id='speakers']/li)

Travel expenses:

=/dl/dd[0] * /dl/dd[1]

PROGRAMMING CONFERENCE 2023

Invited speakers

Name	Email	Organizer
Ada Lovelace	lovelace@rsoc.ac.uk	
Adele Goldberg	adele@xerox.com	TP
Betty Jean Jennings	betty@rand.com	JE
Margaret Hamilton	hamilton@mit.com	JE

Conference budget

Travel cost per speaker:

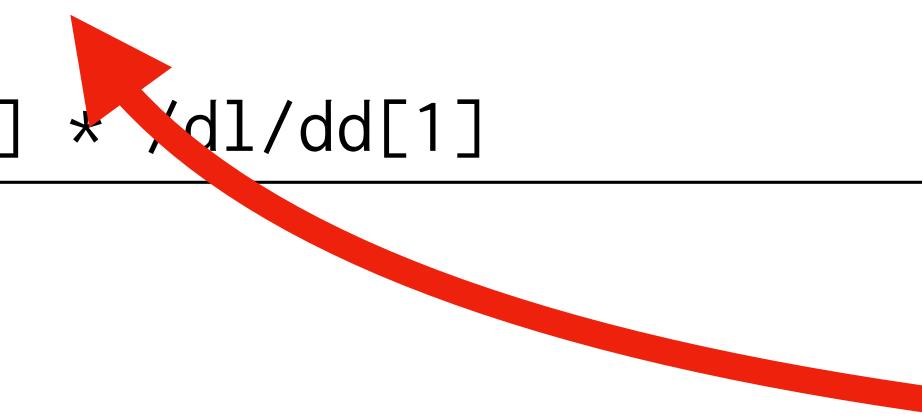
\$1200

Number of speakers:

=COUNT(/table[id='speakers']/tbody/tr)

Travel expenses:

=/dl/dd[0] * /dl/dd[1]



Code refers to structure

Run-time state migration

The Abstract Syntax Definition

```
class Machine  
name: string  
states: State*  
vars: Decl*
```

```
class State  
name: string  
onEntry: Stmt*  
transitions: Trans*
```

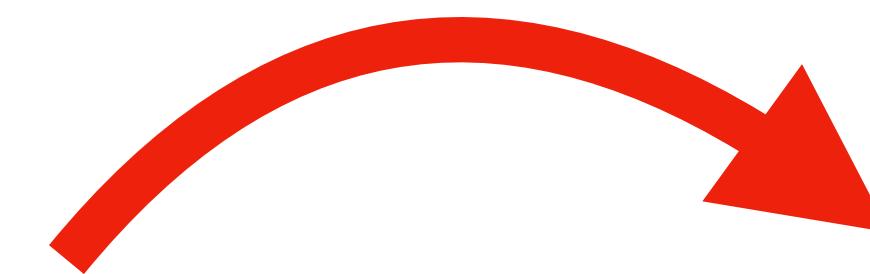
```
class Trans  
event: string  
target: State
```

```
class Decl  
name: string  
type: Type
```

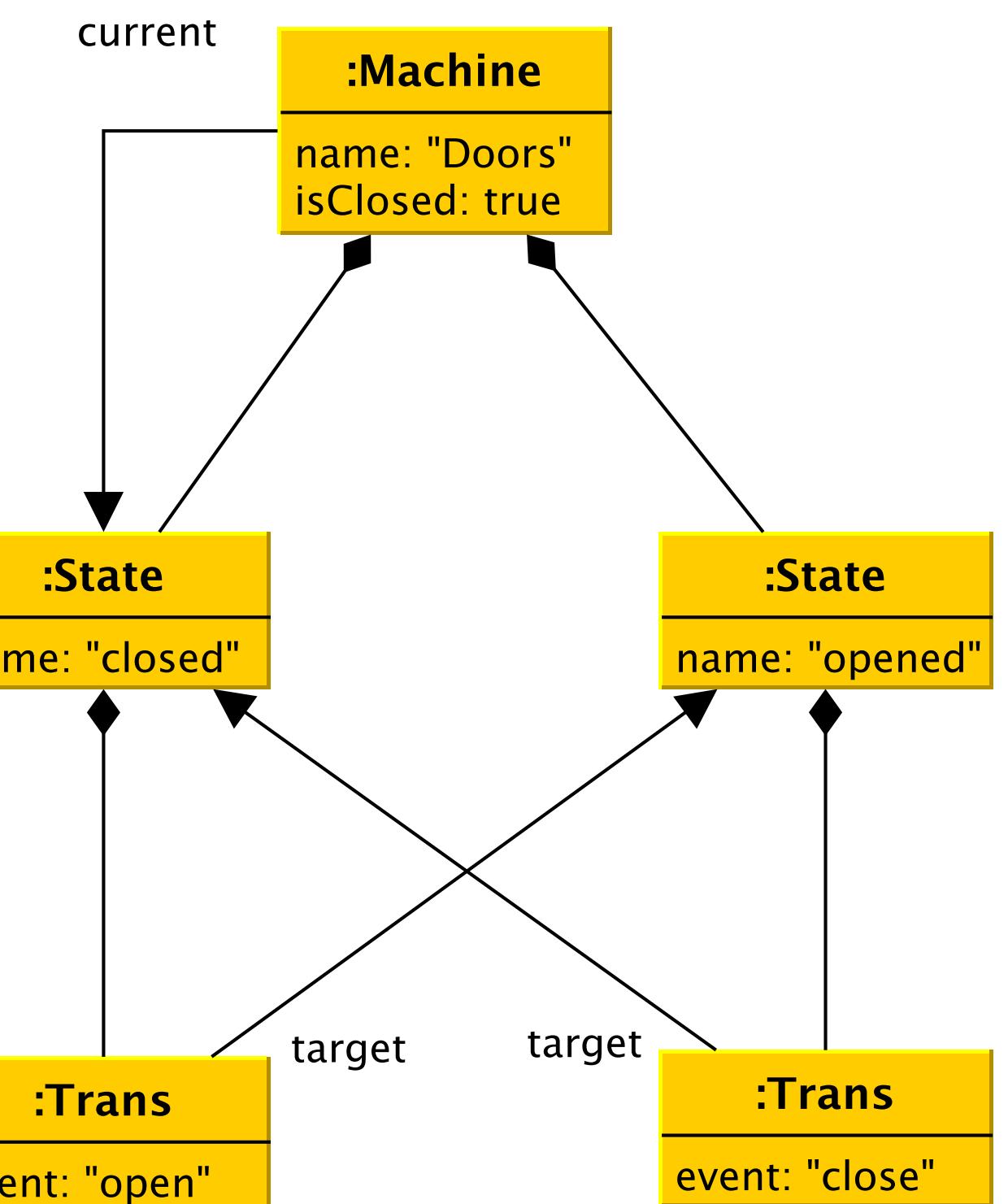
The Program

```
machine Doors  
  
var isClosed: bool = true  
  
state closed  
isClosed := true  
on open => opened  
  
state opened  
isClosed := false  
on close => closed
```

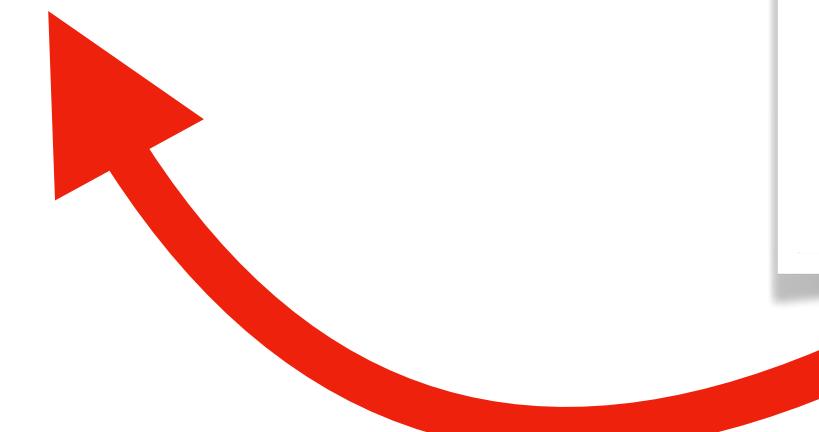
executes as



The Execution



instance of



Run-time state migration

The Abstract Syntax Definition

```
class Machine  
name: string  
states: State*  
vars: Decl*
```

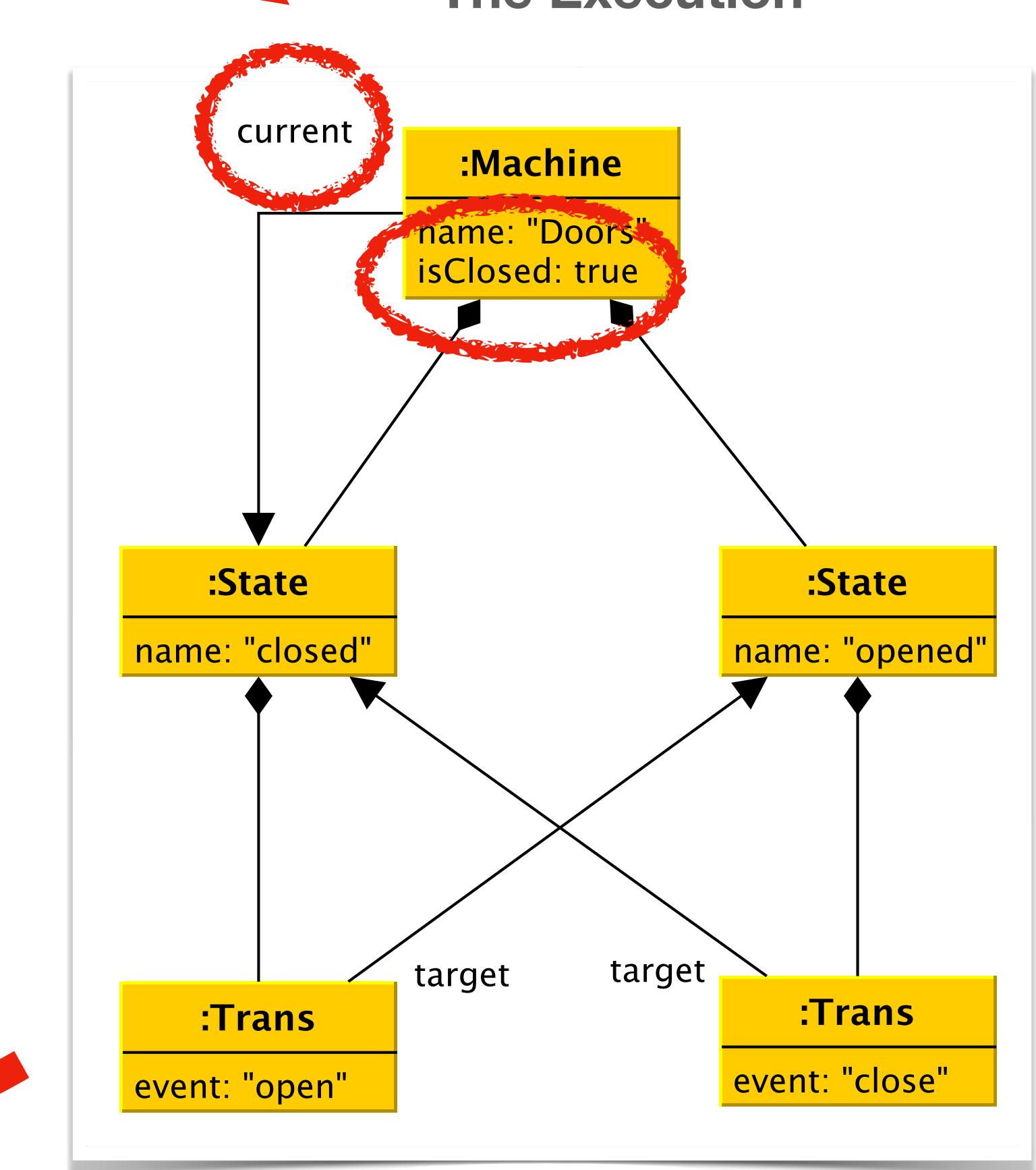
```
class State  
name: string  
onEntry: Stmt*  
transitions: Trans*
```

```
class Trans  
event: string  
target: State  
  
class Decl  
name: string  
type: Type
```

The Program

```
machine Doors  
  
var isClosed: bool = true  
  
state closed  
isClosed := true  
on open => opened  
  
state opened  
isClosed := false  
on close => closed
```

executes as



"extends"

```
class Machine++  
states: State*  
current: State  
isClosed: bool
```

instance of

instance of



Why are these things all the same?

	Meta definition	“Before”	“After”
Database	SQL Schema	denormalized	normalized
Structured Document	(implicit through GUI)	itemized list + sorted	tabular + extra column
Live Modeling	Meta Model	abstract syntax of a program	+ run-time structures
Live Elm	Data Type	list of items	record of field lists

General challenges

- Do minimal harm (less disruptive is better)
- Minimize user intervention
- How to deal with conflicting edits
- No data loss: especially relevant for refactorings
- Heuristics
 - default values: e.g., empty organizer cell
 - reset: e.g., deleting state from state machine if it's active
 - type conversions: e.g., from string to date

Related areas of research:

- *operational transformation*
- *operation-based versioning*
- *CRDTs*
- *database migration*
- *coupled transformations (Lämmel)*
- *lenses*
- *bidirectional trafo community (BX)*
- ...