Reimagining the User Interfaces for Programming

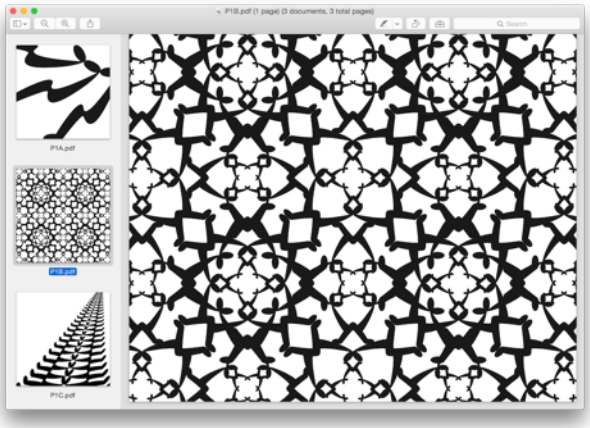
Ravi Chugh

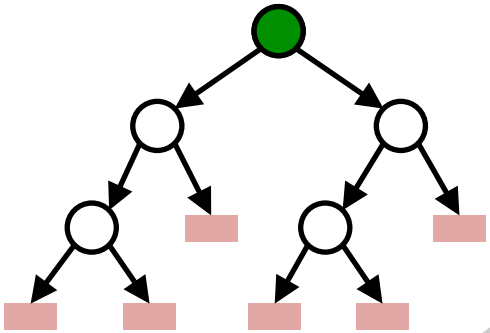
THE UNIVERSITY OF CHICAGO

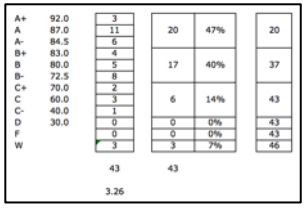
March 2021

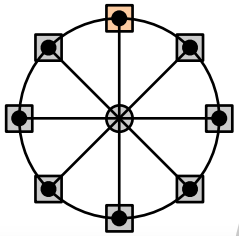
User Interfaces for Programming

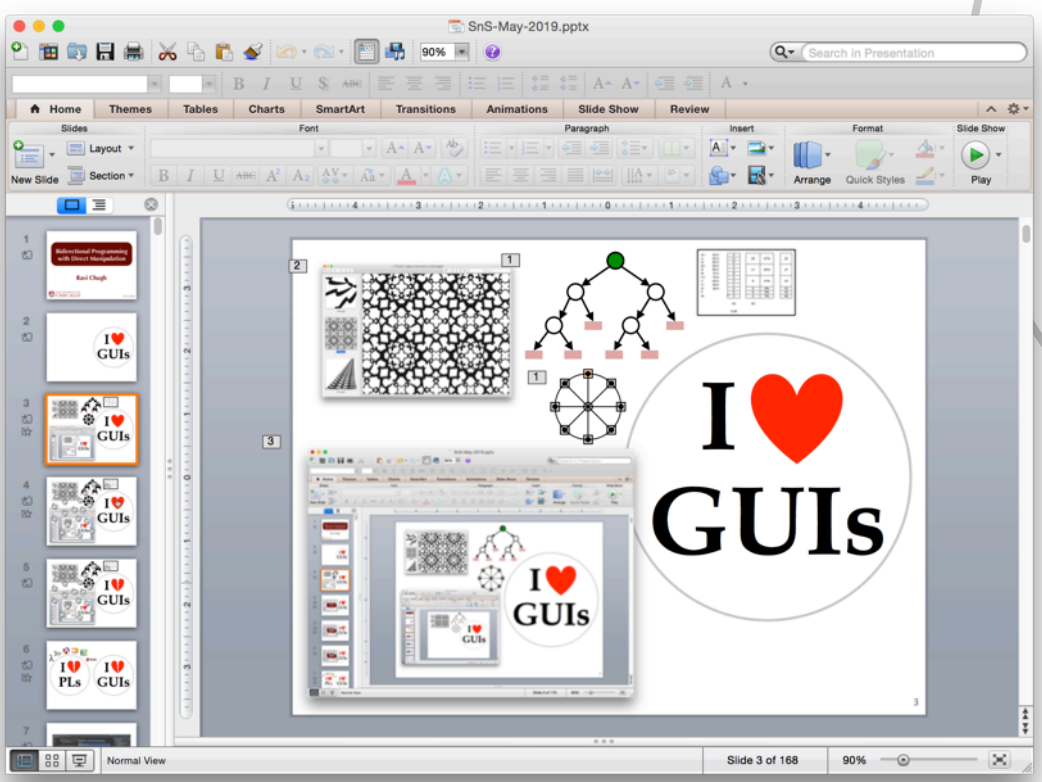
I
GUIs







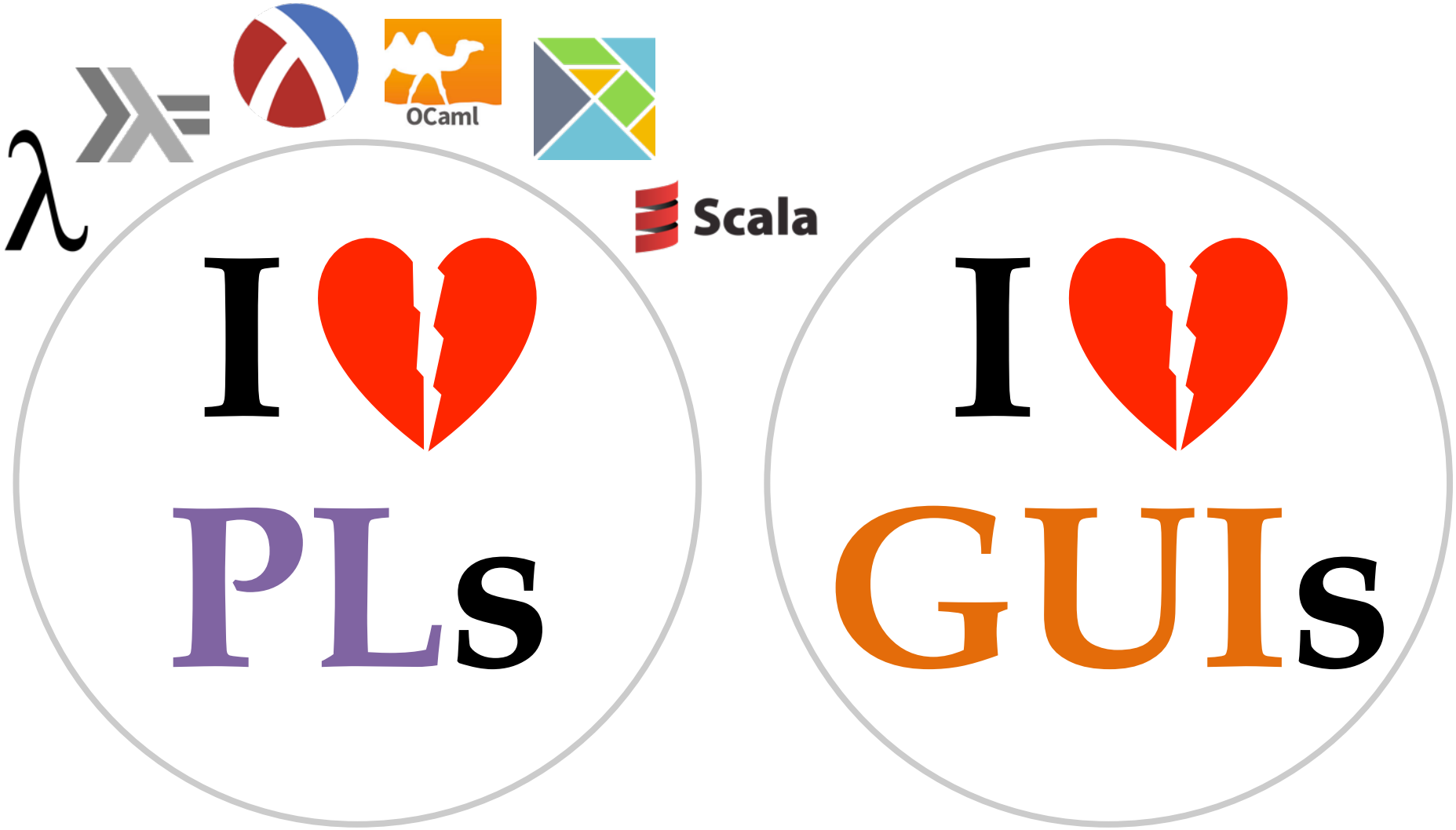


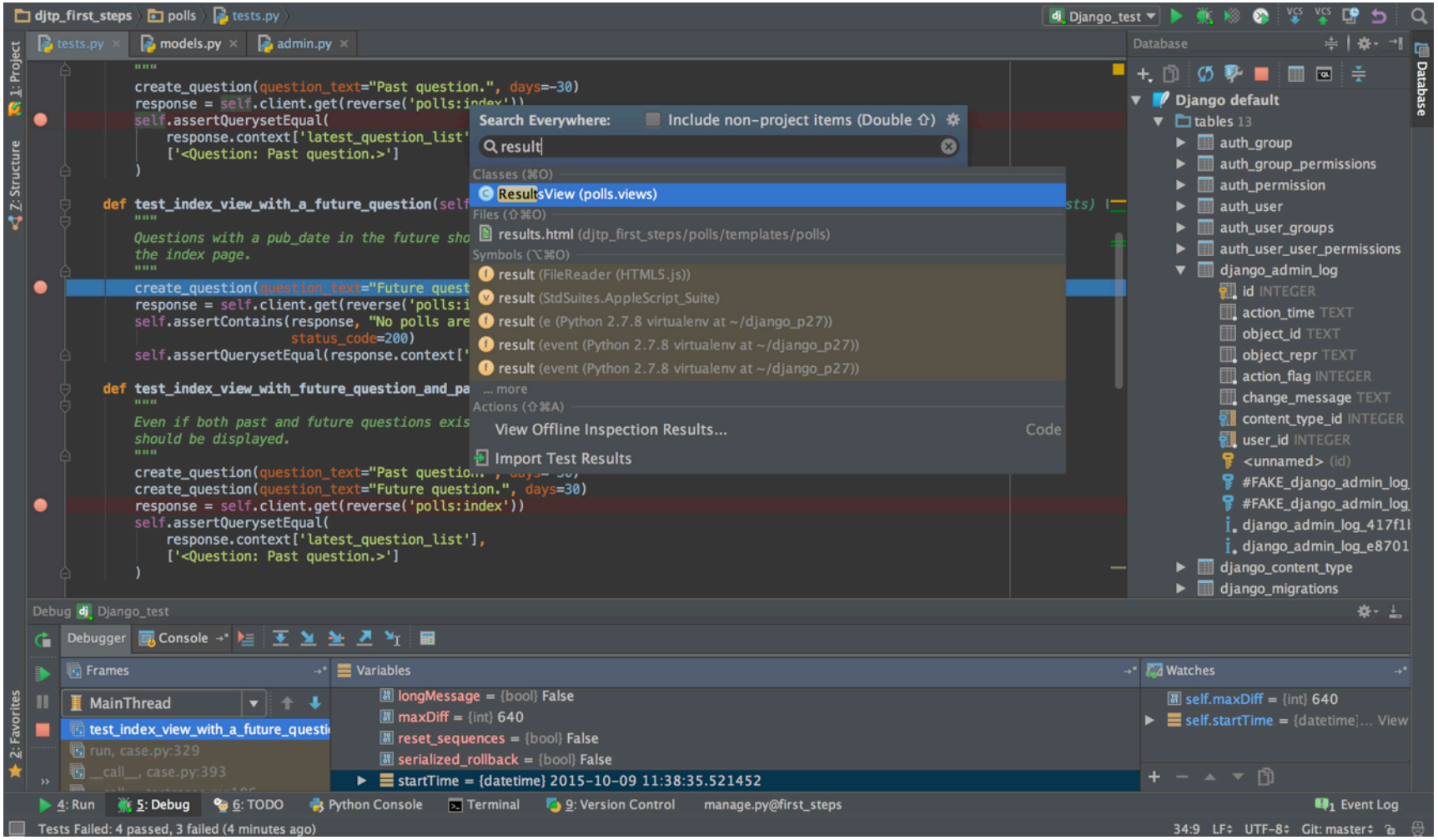

SnS-May-2019.pptx
I ♥ GUIs

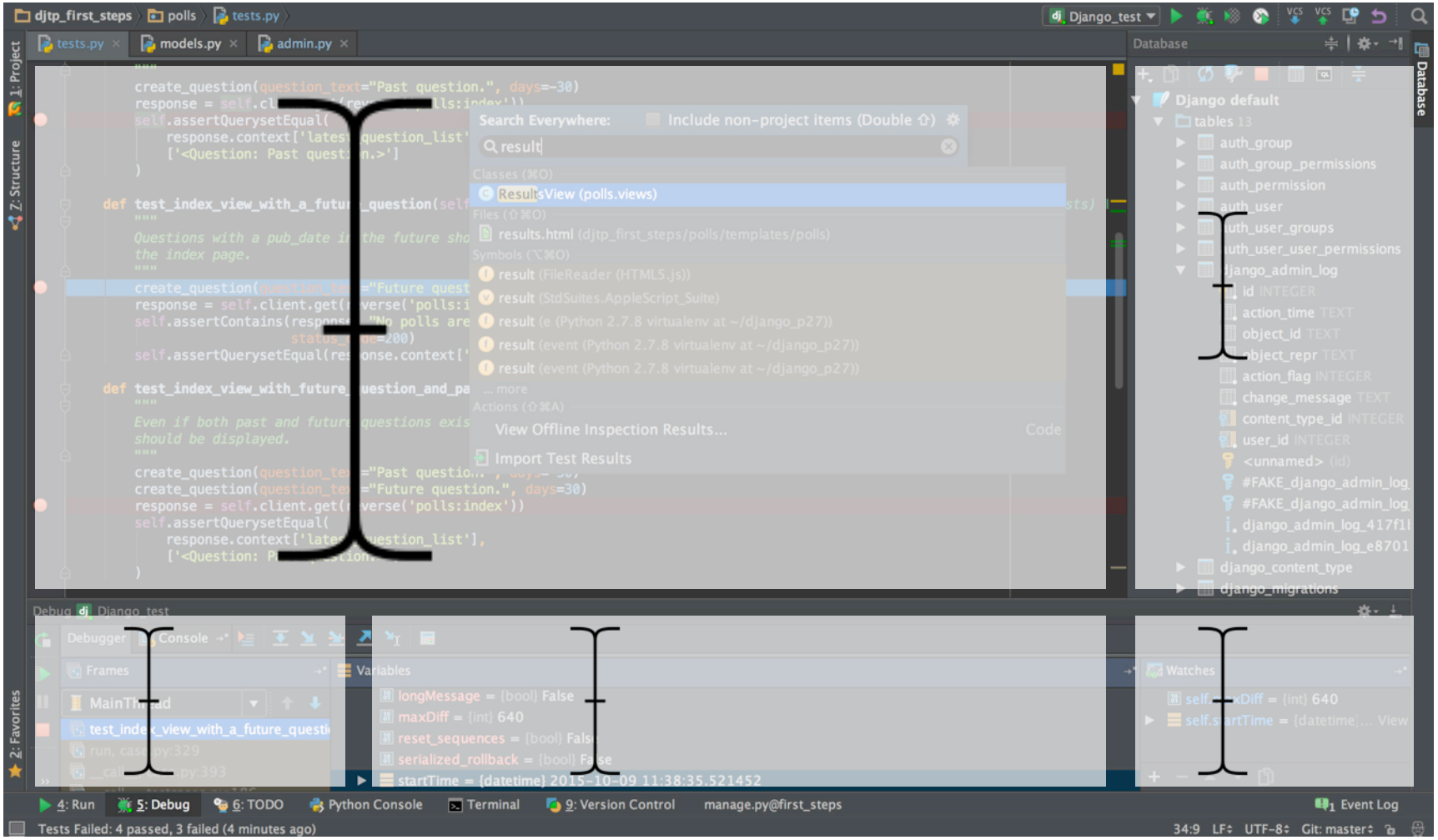
I ♥ GUIs

I
GUIs

I ♥ GUIs


OCaml
Scala
I
PLs
I
GUIs


djtp_first_steps
polls
tests.py
Django_test
tests.py
models.py
admin.py
create_question(question_text="Past question.", days=-30)
response = self.client.get(reverse('polls:index'))
self.assertQuerysetEqual(
response.context['latest_question_list'
['<Question: Past question.>']
)
def test_index_view_with_a_future_question(self
Questions with a pub_date in the future sho
the index page.
create_question(question_text="Future quest
response = self.client.get(reverse('polls:i
self.assertContains(response, "No polls are
status_code=200)
self.assertQuerysetEqual(response.context['
def test_index_view_with_future_question_and_pa
Even if both past and future questions exis
should be displayed.
create_question(question_text="Past question.", days=30)
create_question(question_text="Future question.", days=30)
response = self.client.get(reverse('polls:index'))
self.assertQuerysetEqual(
response.context['latest_question_list'],
['<Question: Past question.>']
)
Search Everywhere:
Include non-project items (Double ⇧)
result
Classes (⌘O)
ResultsView (polls.views)
Files (⇧⌘O)
results.html (djtp_first_steps/polls/templates/polls)
Symbols (⌥⌘O)
result (FileReader (HTML5.js))
result (StdSuites.AppleScript_Suite)
result (e (Python 2.7.8 virtualenv at ~/django_p27))
result (event (Python 2.7.8 virtualenv at ~/django_p27))
result (event (Python 2.7.8 virtualenv at ~/django_p27))
... more
Actions (⇧⌘A)
View Offline Inspection Results...
Code
Import Test Results
Database
Django default
tables 13
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
id INTEGER
action_time TEXT
object_id TEXT
object_repr TEXT
action_flag INTEGER
change_message TEXT
content_type_id INTEGER
user_id INTEGER
<unnamed> (id)
#FAKE_django_admin_log
#FAKE_django_admin_log
django_admin_log_417f1
django_admin_log_e8701
django_content_type
django_migrations
Debug Django_test
Debugger
Console
Frames
Variables
Watches
MainThread
test_index_view_with_a_future_questi
run, case.py:329
__call__, case.py:393
longMessage = {bool} False
maxDiff = {int} 640
reset_sequences = {bool} False
serialized_rollback = {bool} False
startTime = {datetime} 2015-10-09 11:38:35.521452
self.maxDiff = {int} 640
self.startTime = {datetime}... View
4: Run
5: Debug
6: TODO
Python Console
Terminal
9: Version Control
manage.py@first_steps
1 Event Log
Tests Failed: 4 passed, 3 failed (4 minutes ago)
34:9
LF
UTF-8
Git: master


djtp_first_steps
polls
tests.py
Django_test
tests.py
models.py
admin.py
Database
Search Everywhere:
Include non-project items (Double ⇧)
result
Classes (⌘O)
ResultsView (polls.views)
Files (⇧⌘O)
results.html (djtp_first_steps/polls/templates/polls)
Symbols (⌥⌘O)
result (FileReader (HTML5.js))
result (StdSuites.AppleScript_Suite)
result (e (Python 2.7.8 virtualenv at ~/django_p27))
result (event (Python 2.7.8 virtualenv at ~/django_p27))
result (event (Python 2.7.8 virtualenv at ~/django_p27))
... more
Actions (⇧⌘A)
View Offline Inspection Results...
Code
Import Test Results
Debug Django_test
Debugger
Console
Frames
Variables
Watches
longMessage = {bool} False
maxDiff = {int} 640
reset_sequences = {bool} False
serialized_rollback = {bool} False
startTime = {datetime} 2015-10-09 11:38:35.521452
Django default
tables 13
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
id INTEGER
action_time TEXT
object_id TEXT
object_repr TEXT
action_flag INTEGER
change_message TEXT
content_type_id INTEGER
user_id INTEGER
<unnamed> (id)
#FAKE_django_admin_log
#FAKE_django_admin_log
django_admin_log_417f1
django_admin_log_e8701
django_content_type
4: Run
5: Debug
6: TODO
Python Console
Terminal
9: Version Control
manage.py@first_steps
1 Event Log
Tests Failed: 4 passed, 3 failed (4 minutes ago)
34:9
LF
UTF-8
Git: master

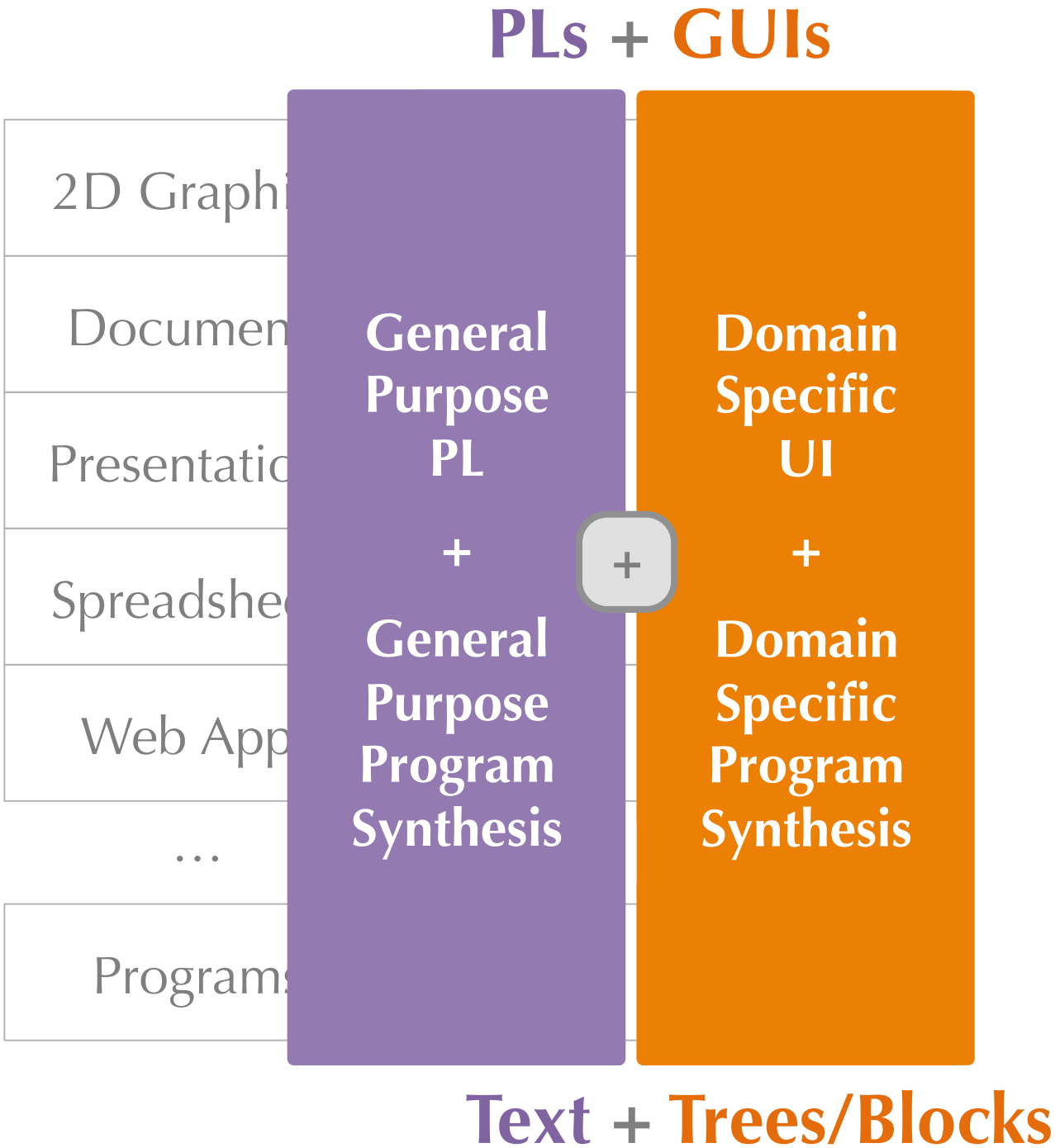
I
PLs

I
GUIs

	PLs	GUIs
2D Graphics		
Documents	LaTeX, Scholarly Markdown	
Presentations	LaTeX	
Spreadsheets	D3, SQL	
Web Apps	JS	

...

	PLs	GUIs
2D Graphics		
Documents		
Presentations		
Spreadsheets		
Web Apps		
…		
Programs		
	Text	Trees/Blocks


PLs + GUIs
2D Graphi
Documen
Presentatic
Spreadshe
Web App
...
Program
General Purpose PL
+
General Purpose Program Synthesis
+
Domain Specific UI
+
Domain Specific Program Synthesis
Text + Trees/Blocks

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

Sketch-n-Sketch

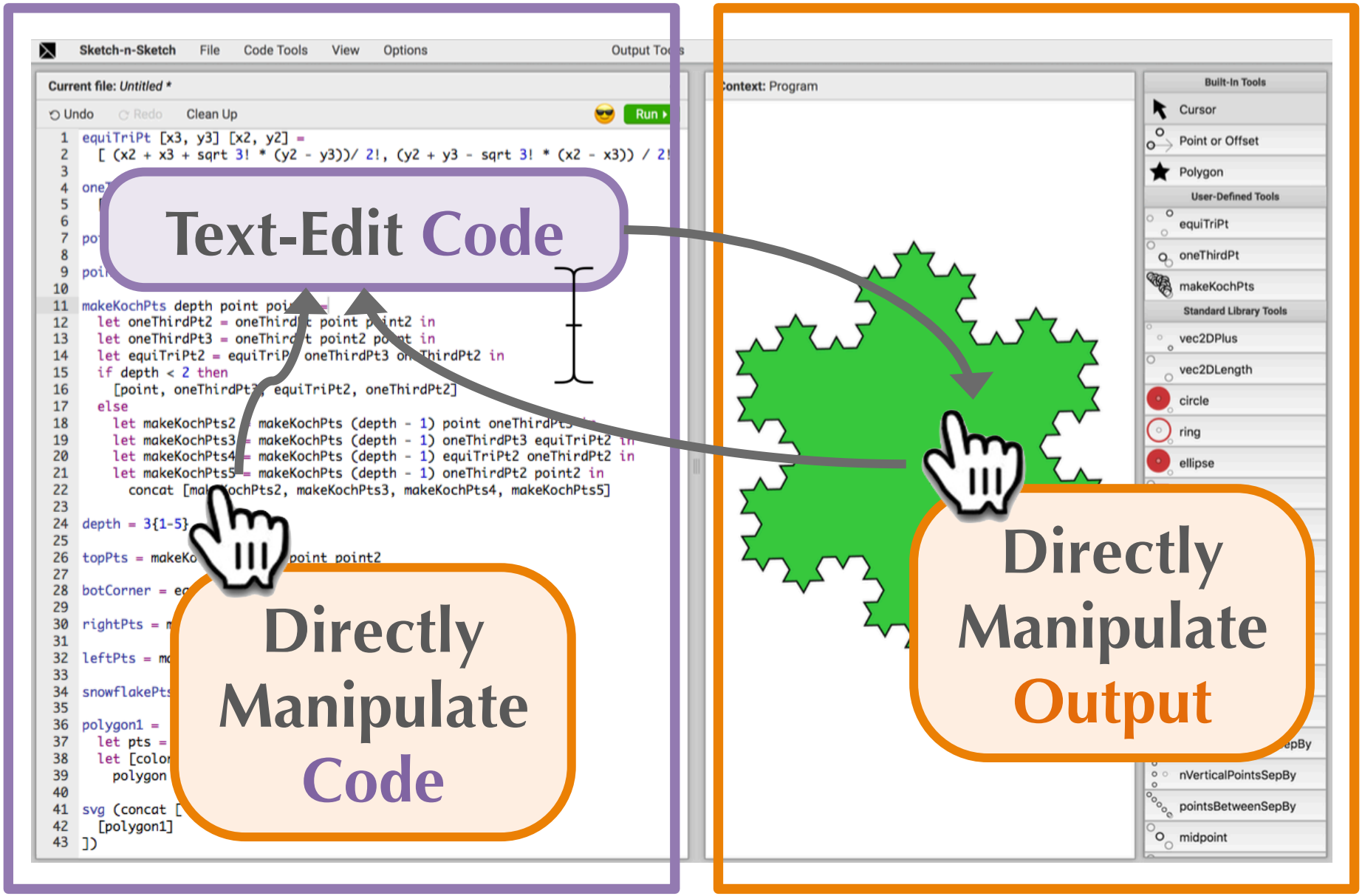
Hybrid Text-GUI Code Editors

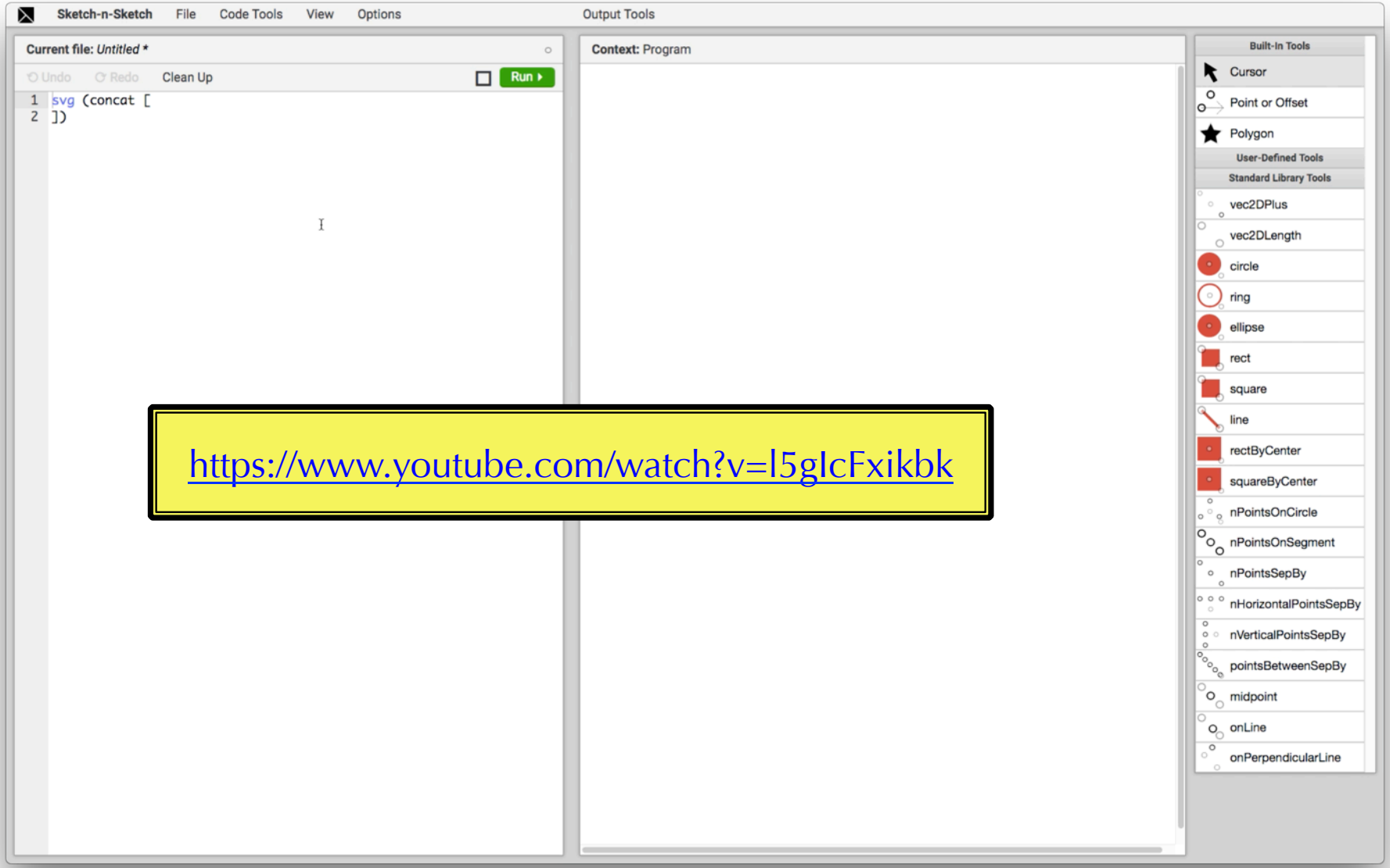
[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Sketch the **Code** + Sketch the **Output**





https://www.youtube.com/watch?v=l5gIcFxikbk

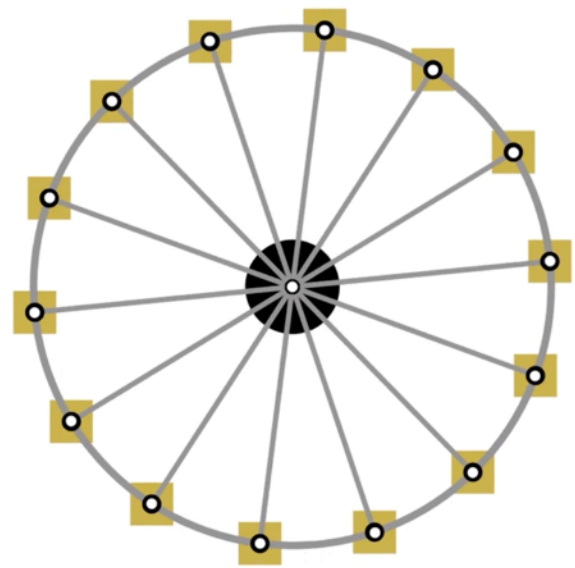
Sketch-n-Sketch File Examples Code Tools Output Tools View Options

Current file: *Untitled ((xi) Ferris Wheel) **

Undo Redo Clean Up Run

```
point = [307, 334]

r = 241

attachmentPts = nPointsOnCircle 14{0-20} 0.12560000000000002{ 3.1

color = 434

spokeFunc point2 =
  line color 5 point point2

spokes =
  map spokeFunc attachmentPts

carFunc center2 =
  squareByCenter 48 center2 20

cars =
  map carFunc attachmentPts

capFunc point2 =
  circle 364 point2 9

caps =
  map capFunc attachmentPts

ring1 = ring color 7 point r

hub = circle 362 point 44

svg (concat [
  [hub],
  cars,
  spokes,
  [ring1],
  caps
])
```

Context: Program



rot: 0.125..

n: 14

Built-In Tools

Cursor

Point or Offset

Polygon

User-Defined Tools

Standard Library Tools

vec2DPlus

vec2DLength

circle

ring

ellipse

rect

square

line

rectByCenter

squareByCenter

nPointsOnCircle

nPointsOnSegment

nPointsSepBy

nHorizontalPointsSepBy

nVerticalPointsSepBy

pointsBetweenSepBy

midpoint

onLine

onPerpendicularLine

https://www.youtube.com/watch?v=6TGBTuuaV3E

Sketch-n-Sketch File View Options ◂ Previous Example Next Example ▸

Current file: Untitled (1a: Table of States)

Undo Redo Run ▸

```
states =
  [ ["Alabama", "AL?", ""]
  , ["Alaska", "AL?", ""]
  , ["Arizona", "AR?", ""]
  , ["Arkansas", "AR?", ""]
  , ["California", "CA", ""]
  , ["Colorado", "CO?", ""]
  , ["Connecticut", "CO?", ""] ]

main =
  let headers = ["State", "Capital"] in
  let rows =
    List.map
      (\[state, abbrev, cap] -> [state, cap + ", " + abbrev])
      states
  in
  let padding = ["padding", "3px"] in
  let headerRow =
    let styles = [padding] in
    Html.tr [] [] (List.map (Html.th styles []) headers)
  in
  let stateRows =
    let colors = ["lightgray", "white"] in
    let drawRow i row =
      let color = List.nth colors (mod i (List.length colors)) in
      let columns =
        List.map
          (Html.
          row
      in
      Html.tr []
    in
    List.indexedMap drawRow rows
  in
  Html.table [padding] [] (headerRow :: stateRows)
```

State	Capital
Alabama	, AL?
Alaska	, AL?
Arizona	, AR?
Arkansas	, AR?
California	, CA
Colorado	, CO?
Connecticut	, CO?

GUI
HTML
Value
Auto Sync

https://www.youtube.com/watch?v=pp6yQPrd6bw

Sketch-n-Sketch File Code Tools View Options Previous Example Next Example

Current file: *Untitled (Get Started)* *

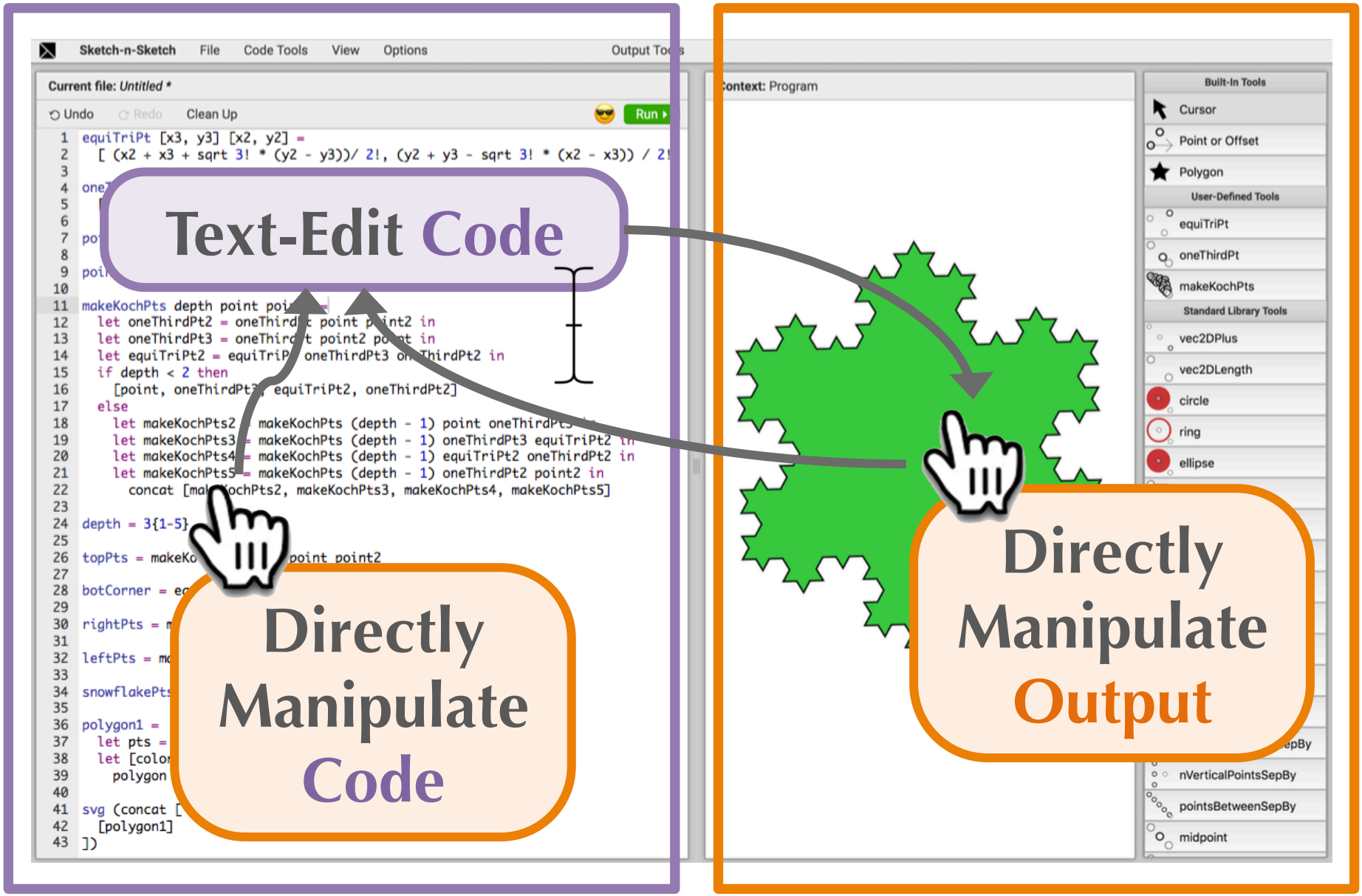
Undo Redo Run

```
(x, y, size) = (23, 30, 217)

rect1 =
  rect "slategray" x y size size

line1 =
  line "white" 10 x y (x + size) (y + size)

line2 =
  line "white" 10 x (y + size)

main = svg [rect1, line1, line2]
```

Raw
Stretchy
Sticky

https://www.youtube.com/watch?v=GE6KSG5Q__0

Sketch the **Code** + Sketch the **Output**



Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Edit Program

Edit Program; Run

Edit Program; Run; View Output

Edit Program; Run; View Output

Edit Program; Run; View Output

Edit Program; Run; View Output

(Edit Program; Run; View Output)+

Edit Program; Run; View Output

Edit Program; Run; View Output

Edit Program; Run; View Output

Edit Program; Run; View Output

Edit Program; Run; View Output

Edit Program; Run; View Output

(Edit Program; Run; View Output; Edit Output)+

Synthesize Program Repair

```
let f x = x ++ " 2018" in
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

$\Rightarrow$

```
"B 2018
@ PLW 2018"
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

⇝

```
"Bx 2019
@ PLW 2018"
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

⇝

```
"Bx 2019
@ PLW 2018"
```

⇐

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

$\Rightarrow$

```
"B 2018
@ PLW 2018"
```

$\rightsquigarrow$

```
"Bx 2019
@ PLW 2018"
```

$\Leftarrow$

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

⇝

```
"Bx 2019
@ PLW 2018"
```

⇐

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

⇒

```
"Bx 2019
@ PLW 2019"
```

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

⇝

```
"Bx 2019
@ PLW 2018"
```

⇐

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

⇒

```
"Bx 2019
@ PLW 2019"
```

Unrestricted function

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

⇝

```
"Bx 2019
@ PLW 2018"
```

⇐

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

⇒

```
"Bx 2019
@ PLW 2019"
```

Unrestricted function

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

Repair "code" + "data"

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

⇐

```
"Bx 2019
@ PLW 2018"
```

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

⇒

```
"Bx 2019
@ PLW 2019"
```

Unrestricted function

```
let f x = x ++ " 2018" in
f "B" ++ " @ " ++ f "PLW"
```

⇒

```
"B 2018
@ PLW 2018"
```

Repair "code" + "data"

```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

⇐

```
"Bx 2019
@ PLW 2018"
```

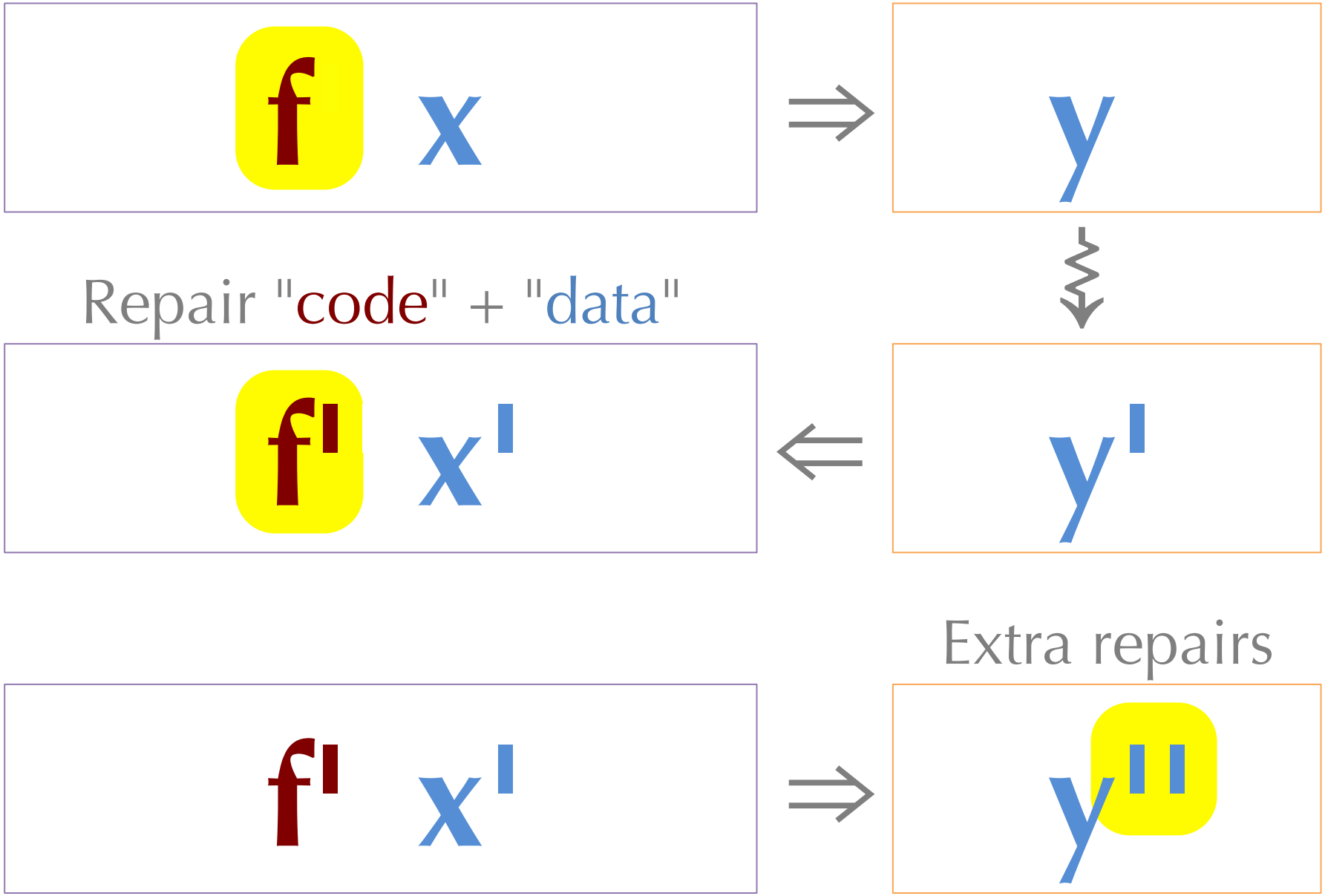
Extra repairs

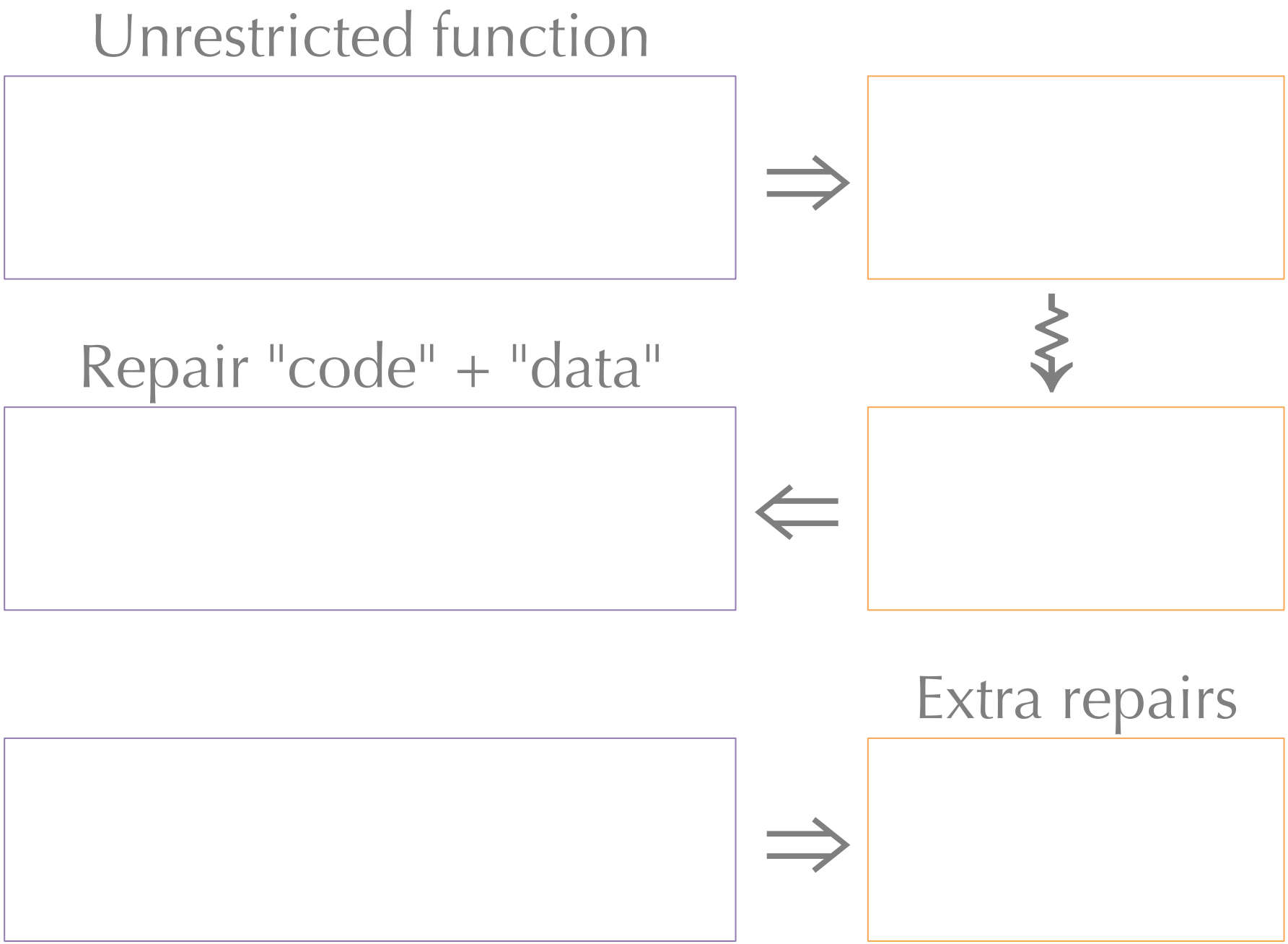
```
let f x = x ++ " 2019" in
f "Bx" ++ " @ " ++ f "PLW"
```

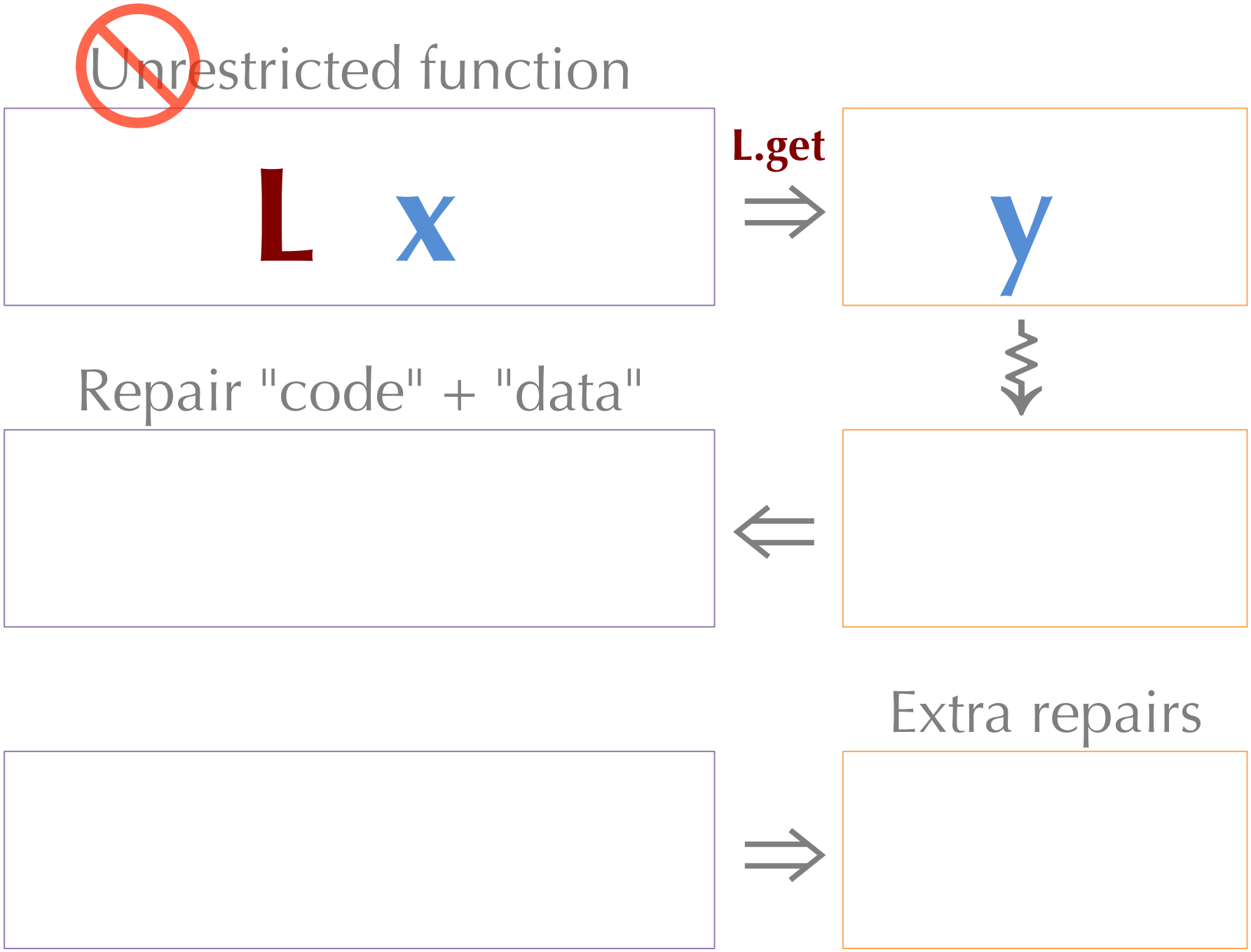
⇒

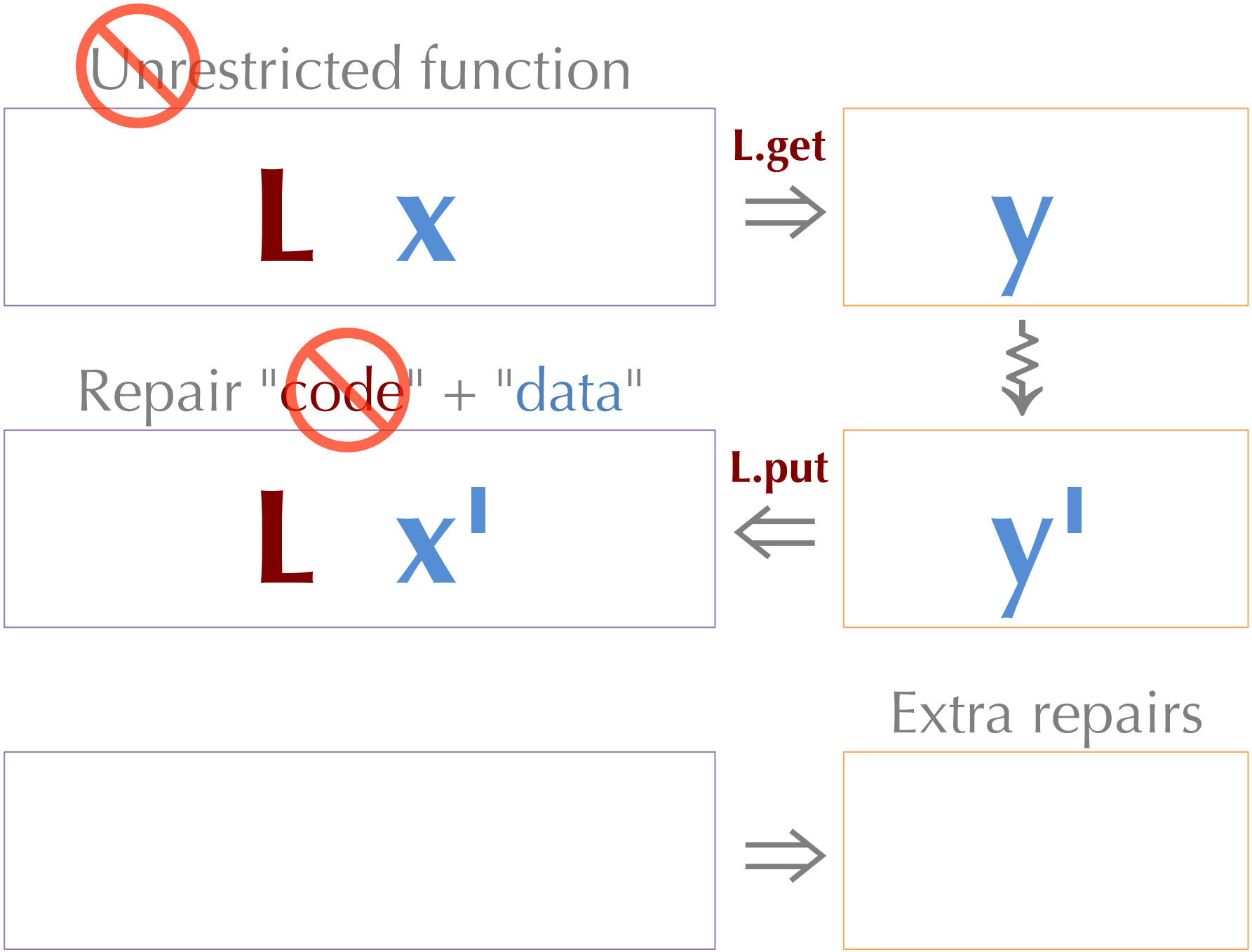
```
"Bx 2019
@ PLW 2019"
```

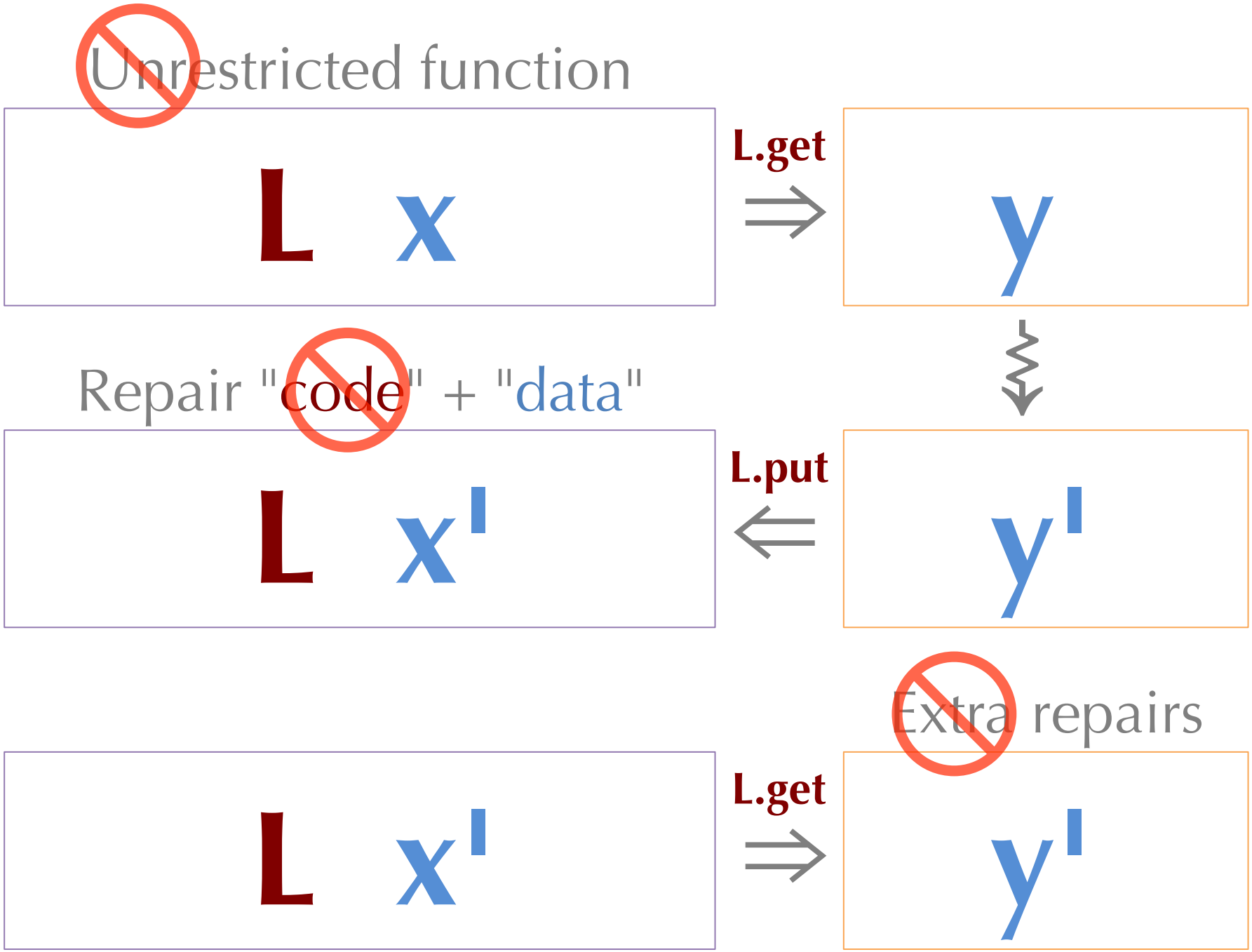
Unrestricted function



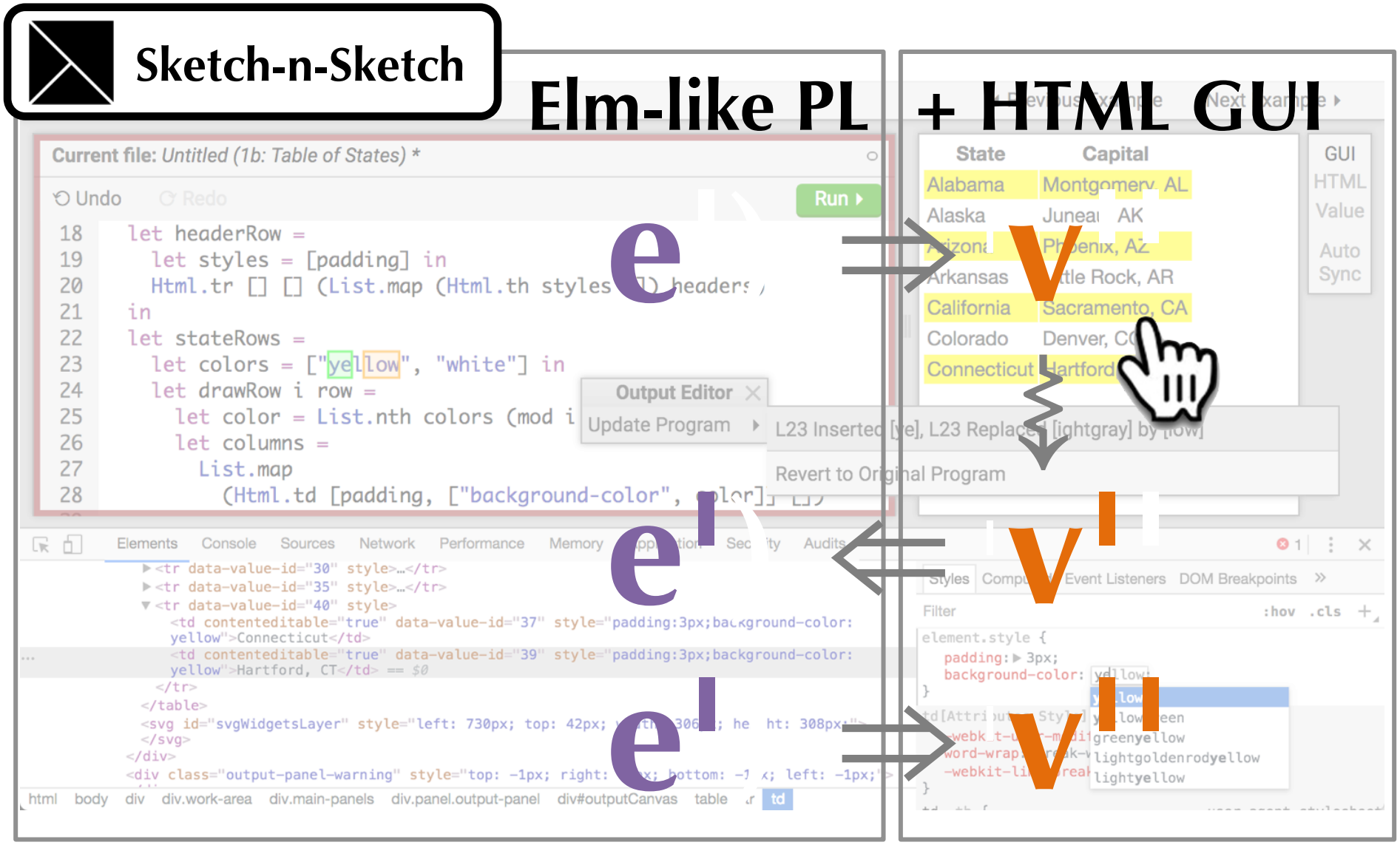

Unrestricted function
Repair "code" + "data"
Extra repairs


Unrestricted function
L x
L.get
y
Repair "code" + "data"
Extra repairs


Unrestricted function
L x
L.get
y
Repair "code" + "data"
L x'
L.put
y'
Extra repairs


Unrestricted function
L x
L.get
y
Repair "code" + "data"
L x'
L.put
y'
Extra repairs
L x'
L.get
y'

Bidirectional Evaluation



Bidirectional Evaluation

$$
\begin{tikzcd}
e \arrow[r, Rightarrow, "\text{eval}"] & v \arrow[d, squiggly] \\
e' & v' \arrow[l, Leftarrow, "\text{uneval}"']
\end{tikzcd}
$$

[>>> Skip to Punchline >>>]

Bidirectional Evaluation

$$(E \vdash e) \overset{\text{eval}}{\Rightarrow} v$$

$$\rightsquigarrow$$

$$(E' \vdash e') \overset{\text{uneval}}{\Leftarrow} v'$$

Environment-style
simpler than
substitution-style

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{tikzcd}
(E \vdash e) \arrow[r, Rightarrow, "\text{eval}"] & v \arrow[d, squiggly] \\
(E' \vdash e') & v' \arrow[l, Leftarrow, "\text{uneval}"']
\end{tikzcd}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & \boxed{c} \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rotatebox{90}{$\rightsquigarrow$} \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{c}
(- \vdash 2018) \Rightarrow 2018 \\
\rightsquigarrow \\
2019
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{c}
(- \vdash 2018) \Rightarrow 2018 \\
\rightsquigarrow \\
(- \vdash 2019) \Leftarrow 2019
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid \boxed{x} \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rotatebox{90}{$\rightsquigarrow$} \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid \boxed{x} \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{c}
(- \vdash \text{let } x = 2018 \text{ in } x) \Rightarrow 2018 \\
\rightsquigarrow \\
2019
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid \boxed{x} \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{c}
((x \mapsto 2018) \vdash x) \Rightarrow 2018 \\
\rightsquigarrow \\
2019
\end{array}
$$

$$e ::= c \mid \lambda x.e \mid \boxed{x} \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$\begin{array}{c} ((x \mapsto 2018) \vdash x) \Rightarrow 2018 \\ \rightsquigarrow \\ ((x \mapsto 2019) \vdash x) \Leftarrow 2019 \end{array}$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid \boxed{x} \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{c}
(- \vdash \text{let } x = 2018 \text{ in } x) \Rightarrow 2018 \\
\rightsquigarrow \\
(- \vdash \text{let } x = 2019 \text{ in } x) \Leftarrow 2019
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rotatebox{90}{$\rightsquigarrow$} \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rightsquigarrow \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$
\begin{array}{c}
(- \vdash \text{let } x = 18 \text{ in } (x, x)) \Rightarrow (18, 18) \\
\rightsquigarrow \\
(19, 18)
\end{array}
$$

$$e ::= c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\boxed{\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}}$$

$$((x \mapsto 18) \vdash (x, x)) \Rightarrow (18, 18)$$

$$\rightsquigarrow$$

$$(19, 18)$$

$$e ::= c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$((x \mapsto 18) \vdash (x, x)) \Rightarrow (18, 18)$$

$$\rightsquigarrow$$

$$((x \mapsto \boxed{\begin{array}{c}18?\\19?\end{array}}) \vdash (x, x)) \Leftarrow (19, 18)$$

$$e ::= c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$((x \mapsto 18) \vdash (x, x)) \Rightarrow (18, 18)$$

$$\rightsquigarrow \qquad \rightsquigarrow$$

$$((x \mapsto \boxed{\begin{array}{c}18?\\19?\\20?\end{array}}) \vdash (x, x)) \Leftarrow (19, \boxed{20})$$

$$
\begin{aligned}
e &::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots \\
v &::= c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E &::= - \mid (E, x \mapsto v)
\end{aligned}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rotatebox{90}{$\rightsquigarrow$} \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

Merge: $E_1 \oplus_{\text{conservative}} E_2$

$(x \mapsto 19) \oplus_{\text{conservative}} (x \mapsto 20)$ undefined

Enables a PutGet theorem

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$
$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$
$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

Merge: $E_1 \oplus_{\text{conservative}} E_2$

$(x \mapsto 19) \oplus_{\text{conservative}} (x \mapsto 20)$ undefined

Enables a PutGet theorem

Merge: $E_1 \oplus_{\text{optimistic}} E_2$

$(x \mapsto 19) \oplus_{\text{optimistic}} (x \mapsto 20) = 19$ or 20 (or 18)

Enables many desirable interactions

$$e ::= c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$((x \mapsto 18) \vdash (x, x)) \Rightarrow (18, 18)$$

$$\rightsquigarrow \quad \rightsquigarrow$$

$$((x \mapsto \begin{array}{c} 18? \\ 19? \\ 20? \end{array}) \vdash (x, x)) \Leftarrow (19, 20)$$

$$e ::= c \mid \lambda x.e \mid x \mid \boxed{(e_1, e_2)} \mid e_1\ e_2 \mid \ldots$$
$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$
$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$(- \vdash \text{let } x = 18 \text{ in } (x, x)) \Rightarrow (18, 18)$$

$$\rightsquigarrow \quad \rightsquigarrow$$

$$(- \vdash \text{let } x = \boxed{\begin{array}{c} 18? \\ 19? \\ 20? \end{array}} \text{ in } (x, x)) \Leftarrow (19, 20)$$

Ask user to choose

$$
\begin{array}{lcl}
e & ::= & c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots \\
v & ::= & c \mid [E]\ \lambda x.e \mid (v_1, v_2) \\
E & ::= & - \mid (E, x \mapsto v)
\end{array}
$$

$$
\begin{array}{c}
(E \vdash e) \Rightarrow v \\
\rotatebox{90}{$\rightsquigarrow$} \\
(E' \vdash e') \Leftarrow v'
\end{array}
$$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$(E \vdash e) \Rightarrow v$$

$$\rightsquigarrow$$

$$(E' \vdash e') \Leftarrow v'$$

$$(- \vdash \begin{array}{l} \text{let } f = \lambda x.\ x \mathbin{++} \text{"18"} \\ \text{in } f\ \text{"B "} \end{array}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$\text{"Bx 19"}$$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$((f \mapsto \lambda x.\ x \mathbin{++} \text{"18"}) \vdash f\ \text{"B "}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$\text{"Bx 19"}$$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\boxed{\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}}$$

$$((x \mapsto \text{"B "}) \vdash x \mathbin{++} \text{"18"}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$\text{"Bx 19"}$$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$((x \mapsto \text{"B "}) \vdash x \mathbin{++} \text{"18"}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$((x \mapsto \text{"B}\boxed{x}\text{"}) \vdash x \mathbin{++} \text{"1}\boxed{9}\text{"}) \Leftarrow \text{"Bx 19"}$$

"data" "code"

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$(E \vdash e) \Rightarrow v$$
$$\rightsquigarrow$$
$$(E' \vdash e') \Leftarrow v'$$

$$((f \mapsto \lambda x.\ x \mathbin{++} \text{"18"}) \vdash f\ \text{"B "}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$((f \mapsto \lambda x.\ x \mathbin{++} \text{"19"}) \vdash f\ \text{"Bx "}) \Leftarrow \text{"Bx 19"}$$

"code" "data"

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid \boxed{e_1\ e_2} \mid \ldots$$

$$v ::= c \mid [E]\ \lambda x.e \mid (v_1, v_2)$$

$$E ::= - \mid (E, x \mapsto v)$$

$$(E \vdash e) \Rightarrow v$$
$$\rightsquigarrow$$
$$(E' \vdash e') \Leftarrow v'$$

$$(- \vdash \text{let } f = \lambda x.\ x \mathbin{++} \text{"18"} \text{ in } f \text{ "B "}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$(- \vdash \text{let } f = \lambda x.\ x \mathbin{++} \text{"19"} \text{ in } f \text{ "Bx "}) \Leftarrow \text{"Bx 19"}$$

"code"

"data"

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rotatebox{90}{$\rightsquigarrow$} \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$(- \vdash \text{"B "} \mathbin{++} \text{"18"}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$\text{"Bx 19"}$$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$

$$\begin{array}{c} (E \vdash e) \Rightarrow v \\ \rightsquigarrow \\ (E' \vdash e') \Leftarrow v' \end{array}$$

$$(- \vdash \text{"B "} \mathbin{++} \text{"18"}) \Rightarrow \text{"B 18"}$$

$$\rightsquigarrow$$

$$(- \vdash \text{"Bx "} \mathbin{++} \text{"19"}) \Leftarrow \text{"Bx 19"}$$

Primitive lens: $((++)_{get}, (++)_{put})$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$
$$\mid \boxed{\text{appLens}\ (e_{get}, e_{put})\ e_x}$$

$$(E \vdash e) \Rightarrow v$$
$$\rightsquigarrow$$
$$(E' \vdash e') \Leftarrow v'$$

Libraries and users can define custom lenses

$$(- \vdash \text{"B "} \mathbin{++} \text{"18"}) \Rightarrow \text{"B 18"}$$
$$\rightsquigarrow$$
$$(- \vdash \text{"Bx "} \mathbin{++} \text{"19"}) \Leftarrow \text{"Bx 19"}$$

Primitive lens: $((++)_{get}, (++)_{put})$

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$
$$\mid \text{appLens}\ (e_{get}, e_{put})\ e_x$$

$$(E \vdash e) \Rightarrow v$$
$$\rightsquigarrow$$
$$(E' \vdash e') \Leftarrow v'$$

Bidirectional Programs are Lenses

$$e ::= c \mid \lambda x.e \mid x \mid (e_1, e_2) \mid e_1\ e_2 \mid \ldots$$
$$\mid \ \mathbf{appLens}\ (e_{get}, e_{put})\ e_x$$

$$(E \vdash e) \Rightarrow v$$
$$\rightsquigarrow$$
$$(E' \vdash e') \Leftarrow v'$$

Bidirectional Programs *with* ~~are~~ Lenses

Example	LOC	Eval	#Upd	#Sol	Average Upd	
States Table A*	37	304±20	11	1.18	85±20	200×
States Table B*	126	774±70	7	1	331±80	700×
Recipe*	193	1455±80	17	1.05	1328±500	16×
Budgetting	37	328±11	7	2	9±2	80×
MVC	71	720±50	10	1	289±80	40×
Linked-Text	91	855±40	5	1.2	2025±200	5×
Markdown	128	1179±110	6	1	1607±200	13×
Dixit	130	705±40	15	1	417±1500	120×
Translation	122	357±20	8	2	415±200	50×
LaTeX in HTML	534	1648±200	6	1	943±1000	150×
Total / **Average**	*1469*	**833±400**	*92*	**1.18**	**(723±900)**	**(70×)**

Example	LOC	Eval	#Upd	#Sol	Average Upd	
States Table A*	37	304±20	11	1.18	85±20	200×
States Table B*	126	774±70	5	[illegible]	332±80	700×
Recipe*	193	1455±80	17	1.05	1328±500	16×
Budgetting	37	328±11	7	2	9±2	80×
MVC	71	720±50	10	1	289±80	40×
Linked-Text	91	855±40	5	1.2	2025±200	5×
Markdown	128	1179±110	6	1	1607±200	13×
Dixit	130	705±40	15	1	417±1500	120×
Translation	122	357±20	8	2	415±200	50×
LaTeX in HTML	534	1648±200	6	1	943±1000	150×
Total / **Average**	*1469*	**833±400**	*92*	**1.18**	**(723±900)**	**(70×)**

Variety of "bidirectional docs"

Example	LOC	Eval	#Upd	#Sol	Average Upd	
States Table A*	37	304±20	11	1.18	85±20	200×
States Table B*	12[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	700×
Recipe*	193	1455±80	17	1.05	1328±500	16×
Budgetting	37[illegible]	[illegible]	[illegible]	[illegible]	9±2	80×
MVC	71	720±50	10	1	289±80	40×
Linked-Text	91	855±40	5	1.2	2025±200	5×
Markdown	128	1179±110	6	1	1607±200	13×
Dixit	130	705±40	15	1	417±1500	120×
Translation	122	357±20	8	2	415±200	50×
LaTeX in HTML	534	1648±200	6	1	943±1000	150×
Total / **Average**	*1469*	**833±400**	*92*	**1.18**	**(723±900)**	**(70×)**

Variety of "bidirectional docs"

Variety of updates with relatively little ambiguity

Example	LOC	Eval	#Upd	#Sol	Average Upd	
States Table A*	37	304±20	11	1.18	85±20	200×
States Table B*	126	774±70	7	1	331±80	700×
Recipe*	193	1455±80	17	1.05	1328±500	16×
Budgetting	37	328±11	5	8	9±2	80×
MVC	71	720±50	10	1	289±80	40×
Linked-Text	91	855±40	5	1.2	2025±200	5×
Markdown	128	1179±110	6	1	1607±200	13×
Dixit	130	705±40	15	1	417±1500	120×
Translation	120	357±20	8	1	415±200	50×
LaTeX in HTML	534	1648±200	6	1	943±1000	150×
Total / **Average**	*1469*	**833±400**	*92*	**1.18**	**(723±900)**	**(70×)**

Variety of "bidirectional docs"

Variety of updates with relatively little ambiguity

Uneval time ≈ Eval time (with optimizations)

[>>> Skip Related Work >>>]

Bidirectional Evaluation

Related Work: Lenses

Lenses [Foster et al. 2007, et al.]

BX DSLs for relational data, trees, strings, graphs

Point-free programming style

Applicative and Monadic Lenses

[Matsuda and Wang, ICFP 2015]
[Xia, Orchard, and Wang, ESOP 2019]

Bidirectional Evaluation

Related Work: Unevaluation

Func. Program Slicing [Perera et al., ICFP 2012]

Forward evaluator records full trace for output value

Backward evaluator slices exp. based on "don't-care" sub-values

HOBiT [Matsuda and Wang, ESOP 2018]

Staged evaluator for λ-calculus eliminates function calls ($\mathbf{e_1}$ $\mathbf{e_2}$)

Bidirectional evaluation for "residual" first-order programs

More direct support for "branch switching" than our approach

Bidirectional Evaluation

Related Work: Expression Provenance

Transparent ML [Acar et al., JCS 2013]

Tracing evaluator records expression provenance

Control-flow choices not recorded

Sketch-n-Sketch [PLDI 2016]

Tracing evaluator records exp. provenance for numeric values

Constraint solving to update numeric values

Bidirectional Evaluation

Work-in-progress: Bidirectional JavaScript

JavaScript ~= λ-calculus + references + label/break + exceptions

[Guha et al., ECOOP 2010]

Unevaluation

Backward rules for each language form

Expression Provenance

Source-to-source rewrite, rather than tracing- or un- evaluator

Domain-Specific Lenses

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

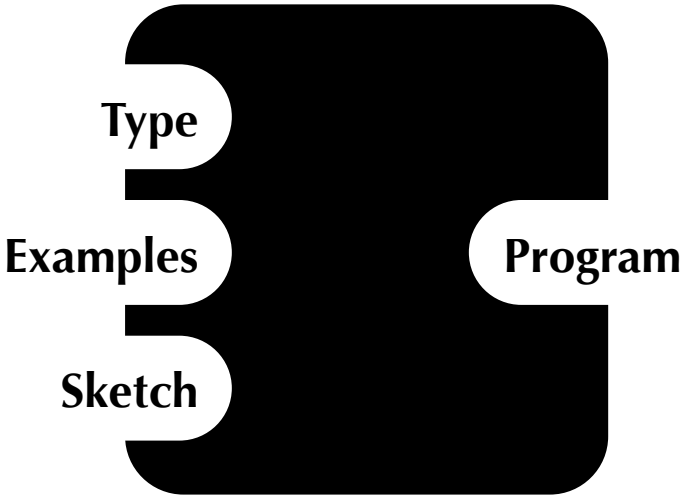
[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

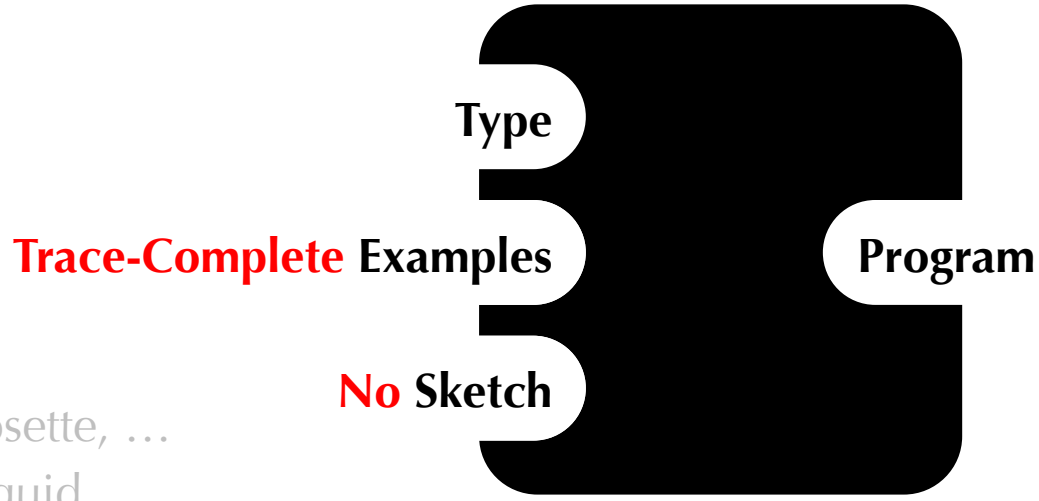
Live Programming by Example

[POPL 2019, ICFP 2020]



Live Programming by Example

[POPL 2019, ICFP 2020]



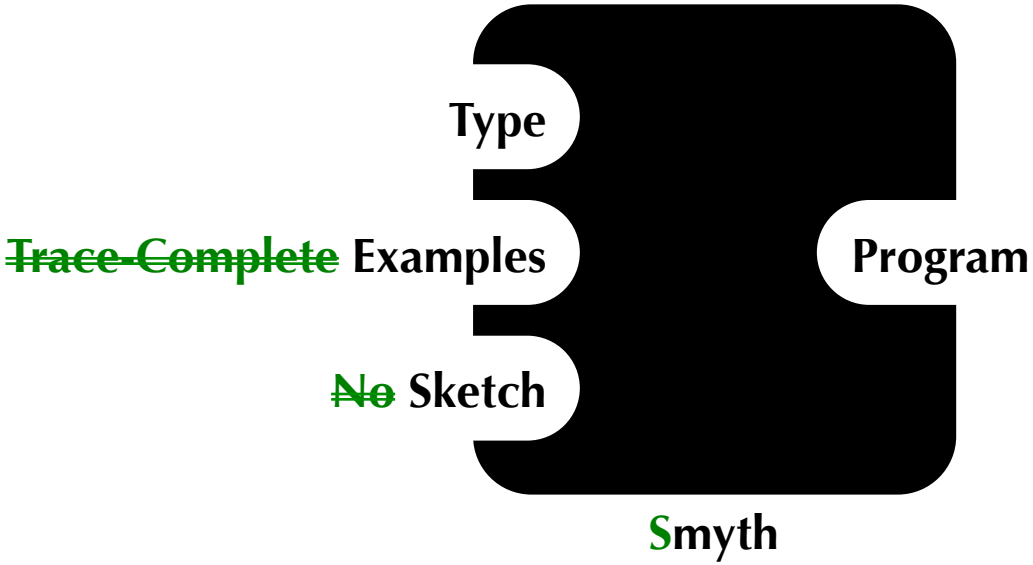
Myth

[Osera et al. PLDI 2015, POPL 2016]

Sketch, Rosette, …
Leon, Synquid, …
FlashFill, PROSE, …

Live Programming by Example

[POPL 2019, ICFP 2020]



Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

PL+UI

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

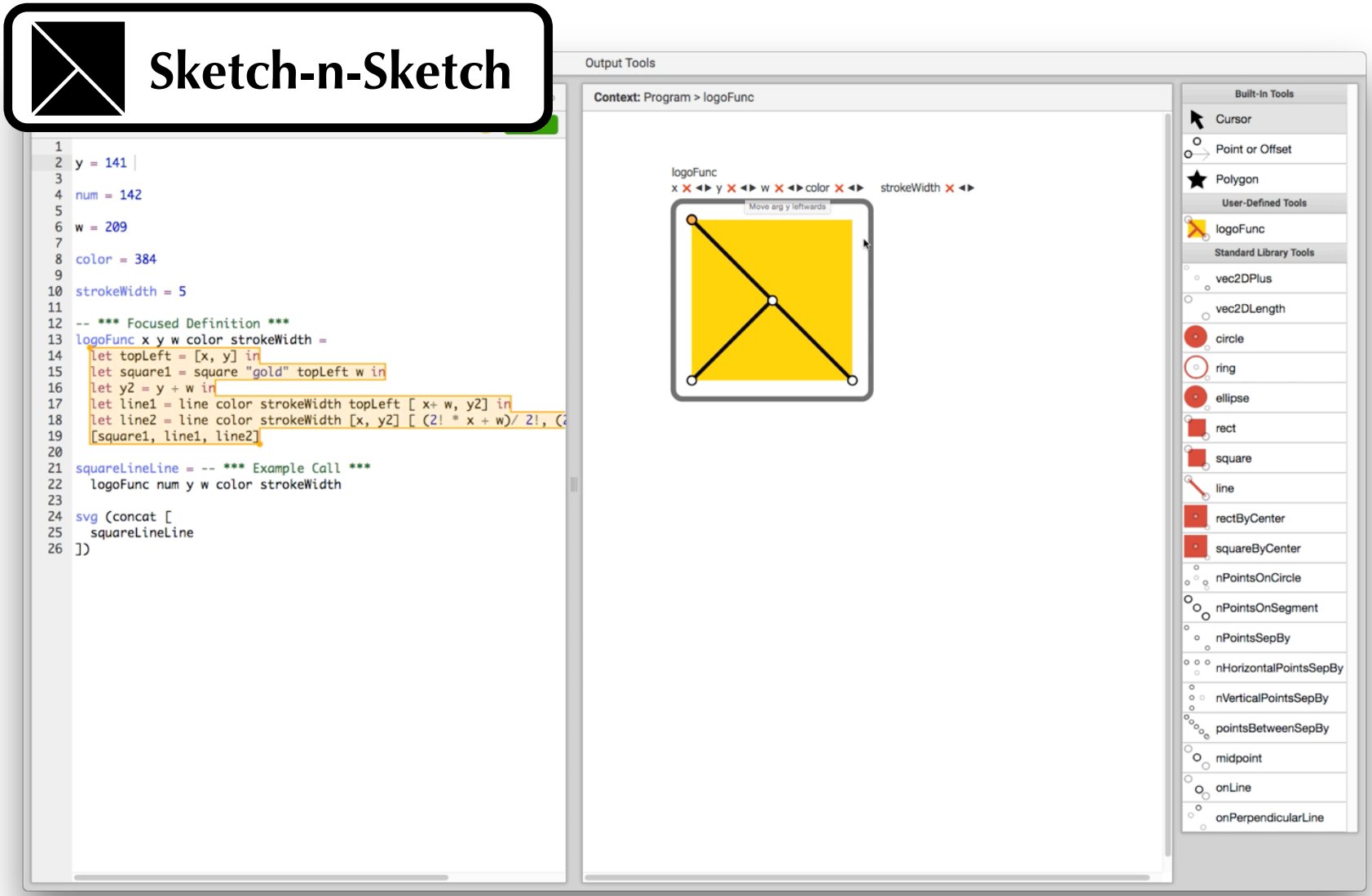
Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

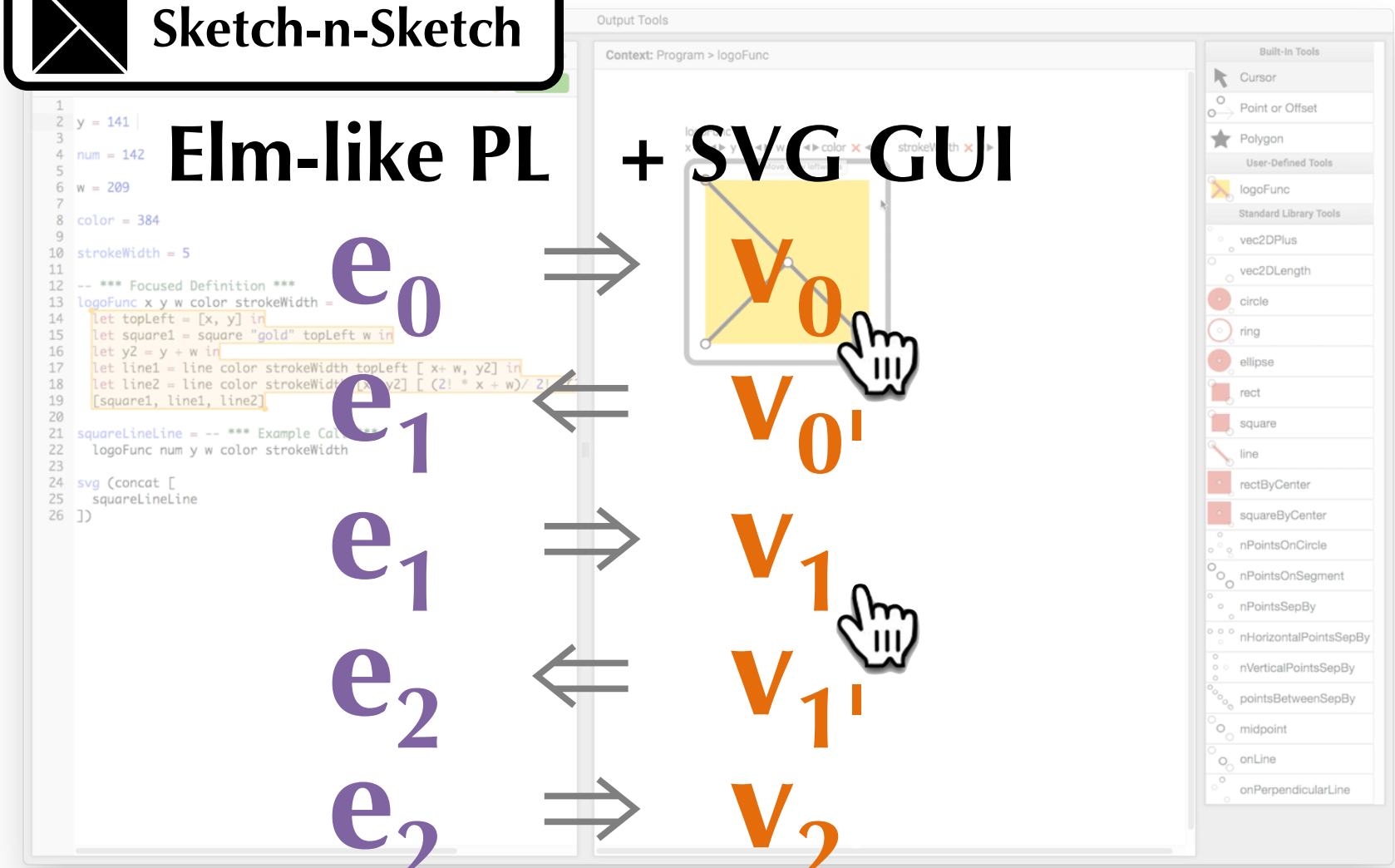
Output-Directed Programming



Output-Directed Programming

Sketch-n-Sketch

Elm-like PL + SVG GUI



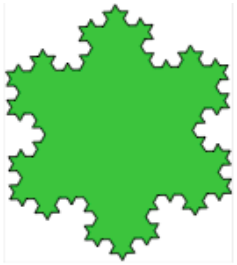
$e_0 \Rightarrow v_0$

$e_1 \Leftarrow v_{0'}$

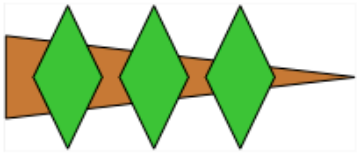
$e_1 \Rightarrow v_1$

$e_2 \Leftarrow v_{1'}$

$e_2 \Rightarrow v_2$



(i) Koch Snowflake ★
31 LOC



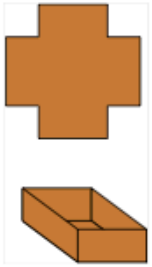
(ii) Tree Branch ★
31 LOC

(iii) Target ★
8 LOC

(iv) Precision Floor Plan ☆
9 LOC, 3 MO
(vs. 25 LOC, 27 MO)

(v) Mondrian Arch λ
15 LOC, 4 MO
(vs. 23 LOC, 9 MO)

(vi) Balance Scale ☆
34 LOC, 21 MO
(vs. 53 LOC, 40 MO)



(vii) Box Volume ★
81 LOC

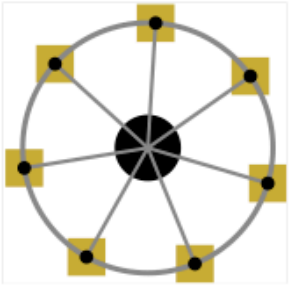
(viii) Xs λ★
44 LOC

(ix) Battery λ
13 LOC, 5 MO
(vs. 21 LOC, 2 MO)

(x) Ladder ★
18 LOC

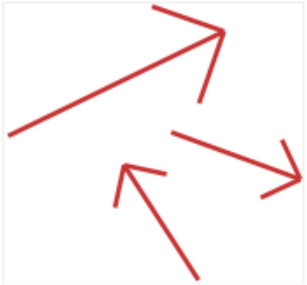
(xi) Logo (via Three Tris) λ★
40 LOC

(xii) N Boxes ★
7 LOC



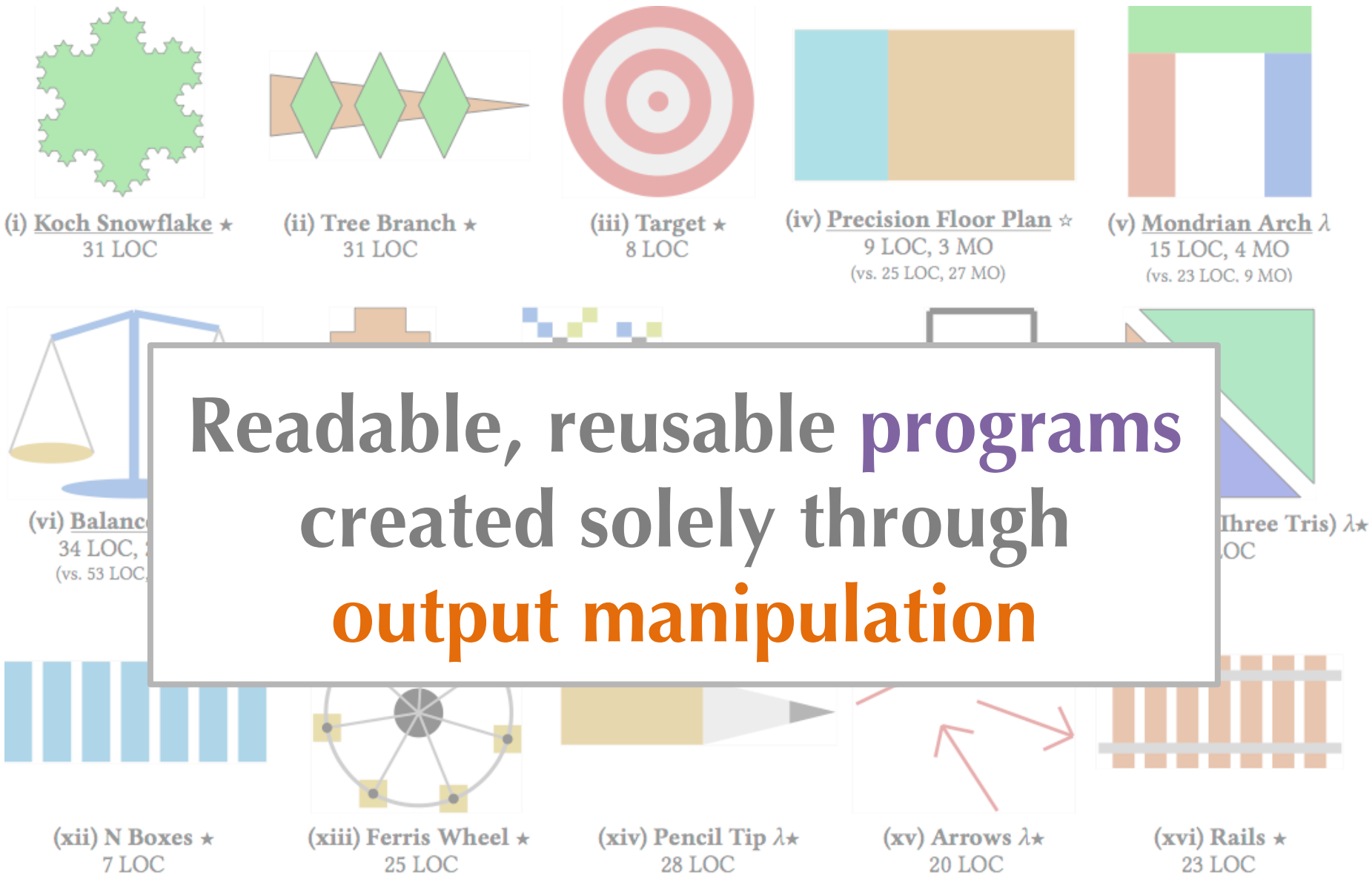
(xiii) Ferris Wheel ★
25 LOC

(xiv) Pencil Tip λ★
28 LOC



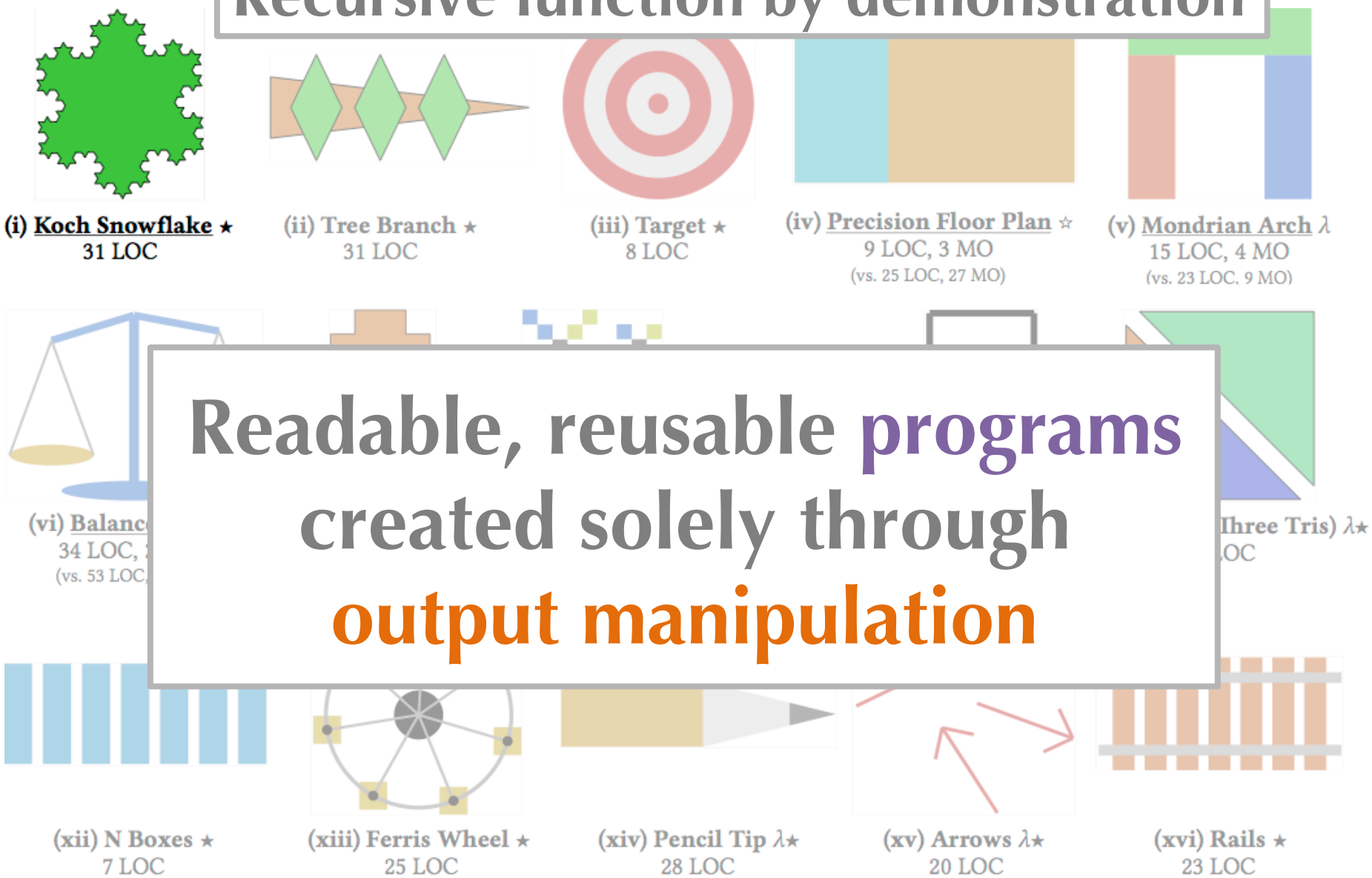
(xv) Arrows λ★
20 LOC

(xvi) Rails ★
23 LOC



Readable, reusable **programs** created solely through **output manipulation**

Recursive function by demonstration



Readable, reusable programs created solely through output manipulation

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

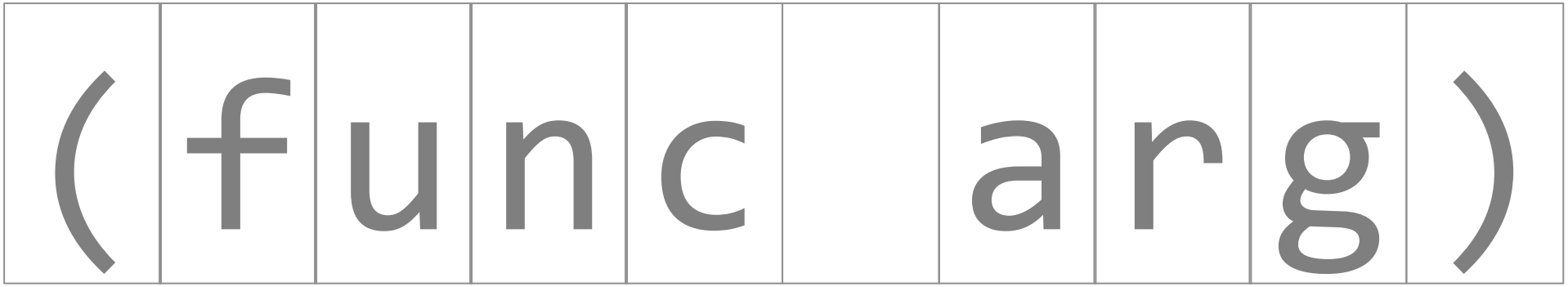
PL + UI

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

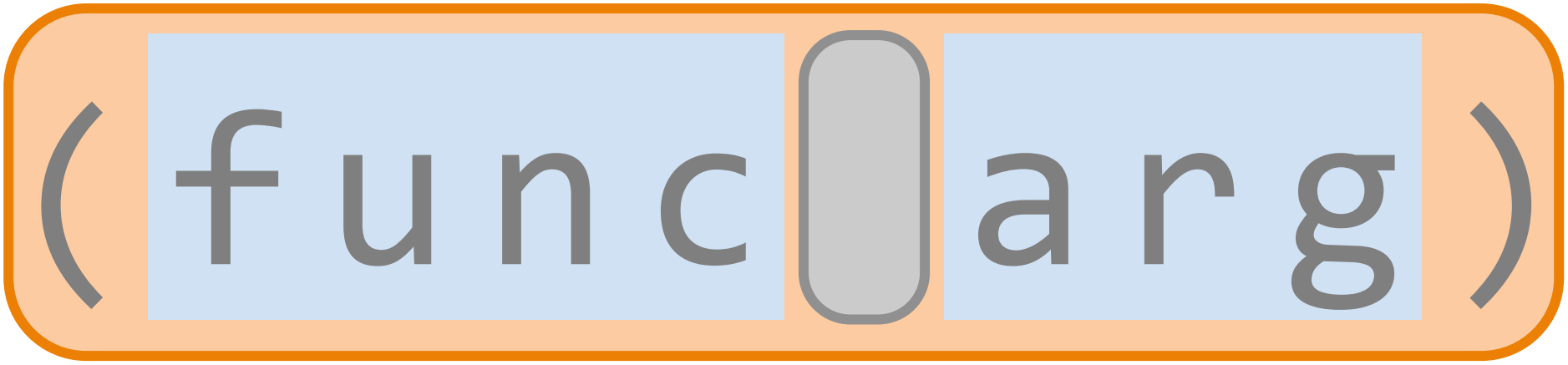
[UIST 2016, UIST 2019]



Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Show AST on Text

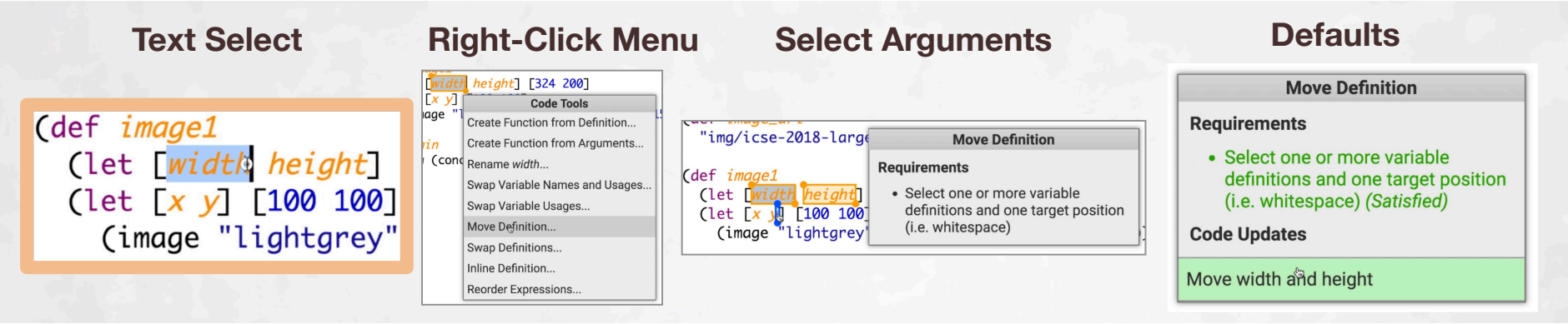


Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

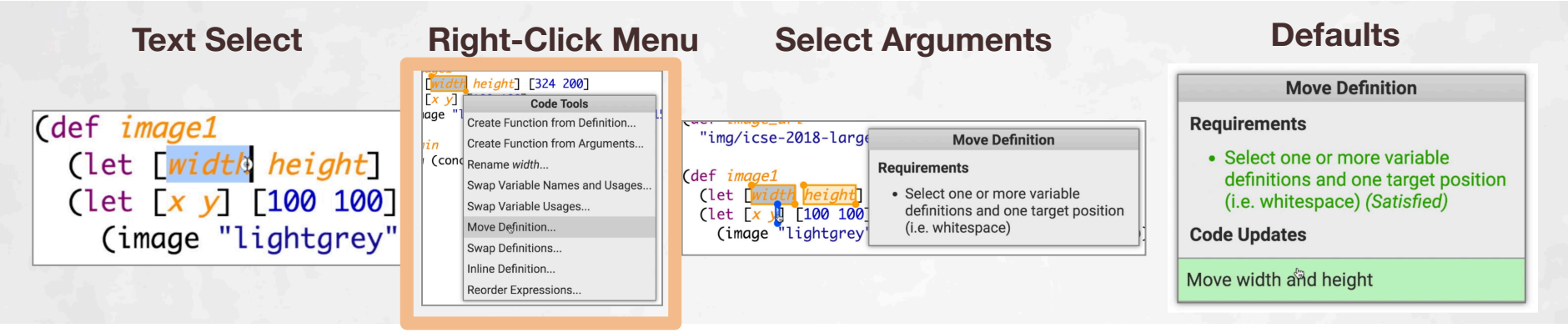
>>> Skip User Study >>>

Traditional Refactoring UI



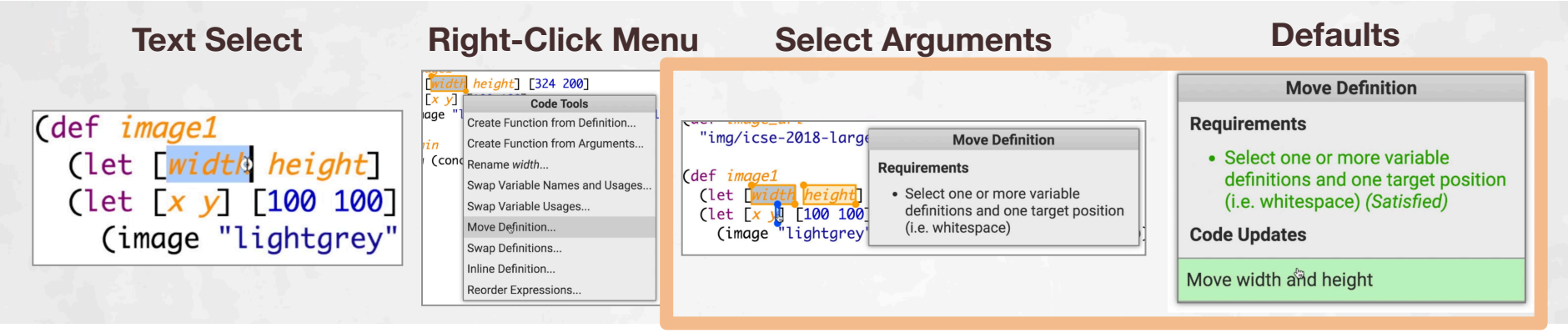
Text-Select

Traditional Refactoring UI



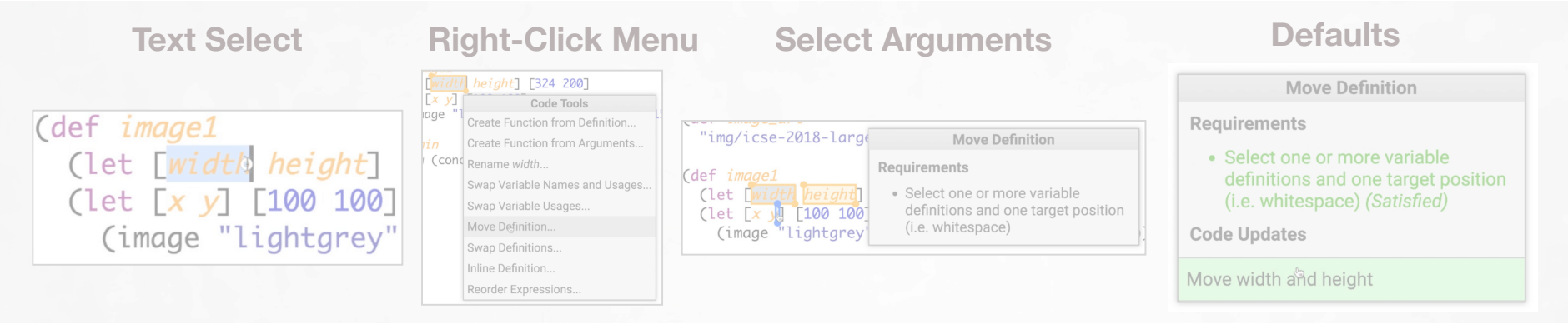
Text-Select + Long Menu

Traditional Refactoring UI



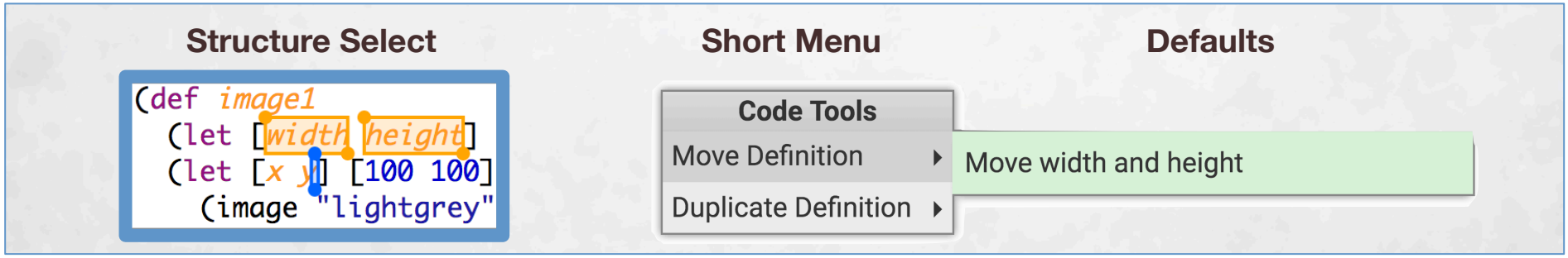
Text-Select + Long Menu + Configuration Choices

Traditional Refactoring UI



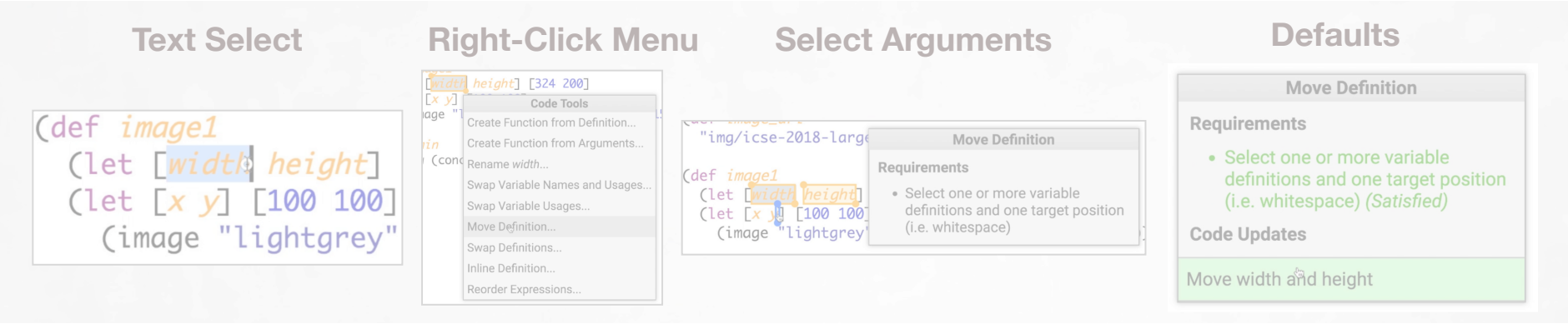
Text-Select + Long Menu + Configuration Choices

Deuce Refactoring UI



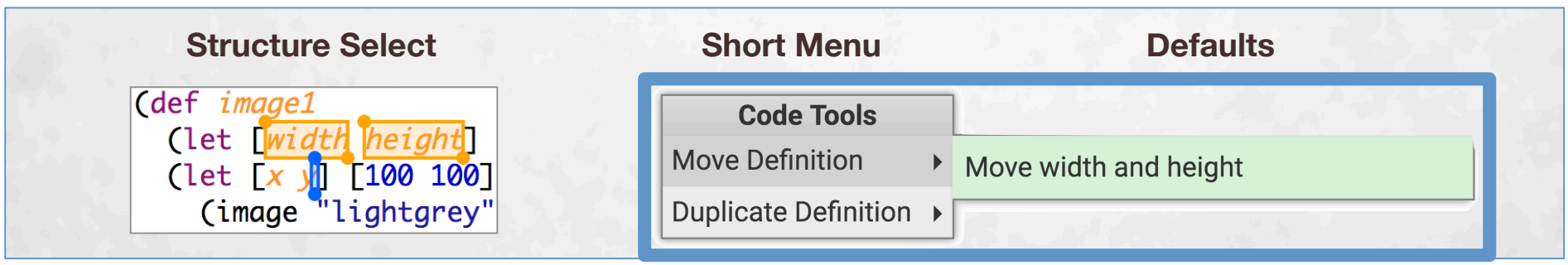
Structural Multi-Select

Traditional Refactoring UI

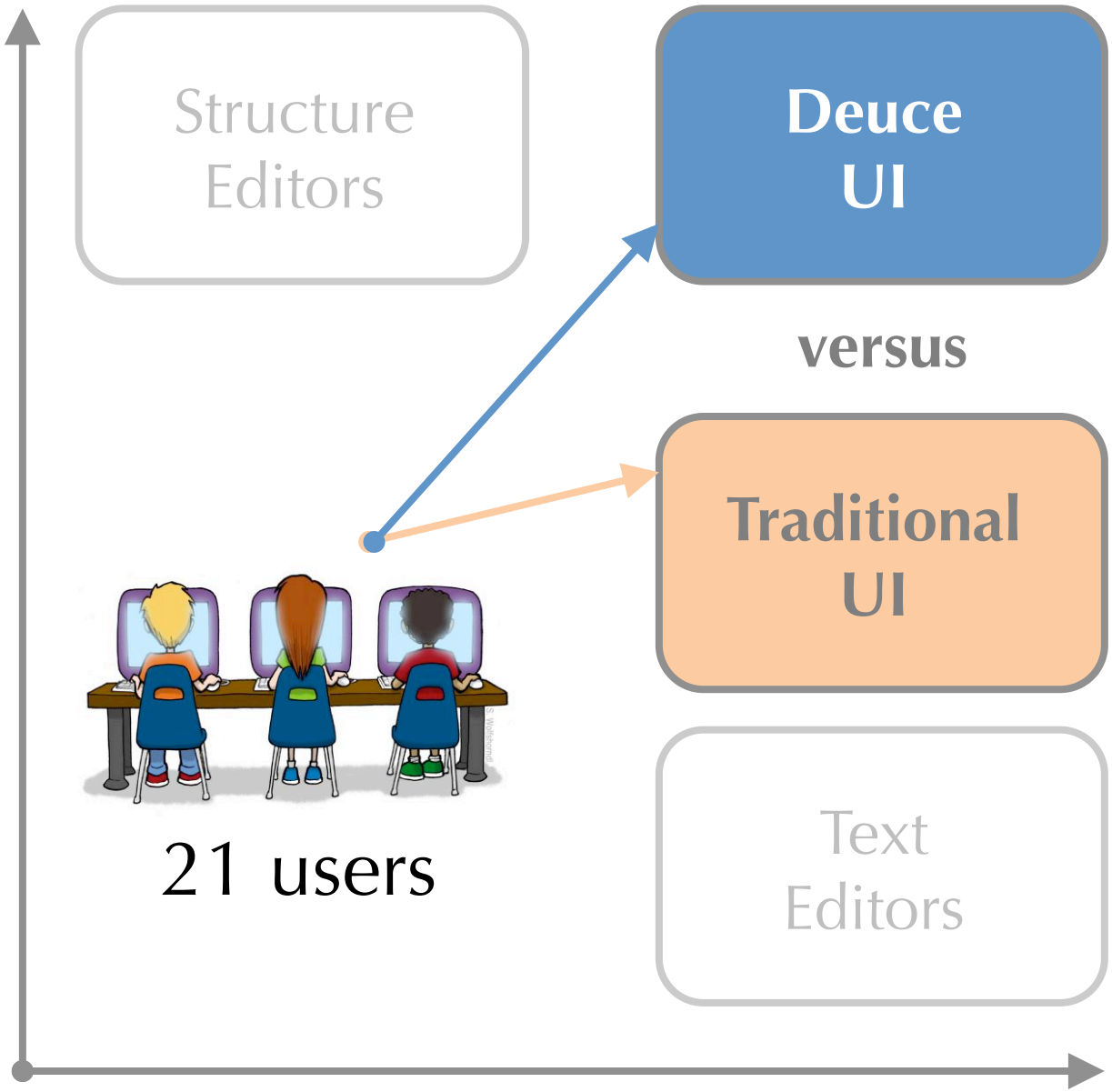


Text-Select + Long Menu + Configuration Choices

Deuce Refactoring UI



Structural Multi-Select + Context-Sensitive Transforms


Structure Editors
Deuce UI
versus
Traditional UI
21 users
Text Editors

Deuce UI versus **Traditional UI**

Traditional may be better for discovery

Deuce may be faster once learned

Deuce strongly preferred

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

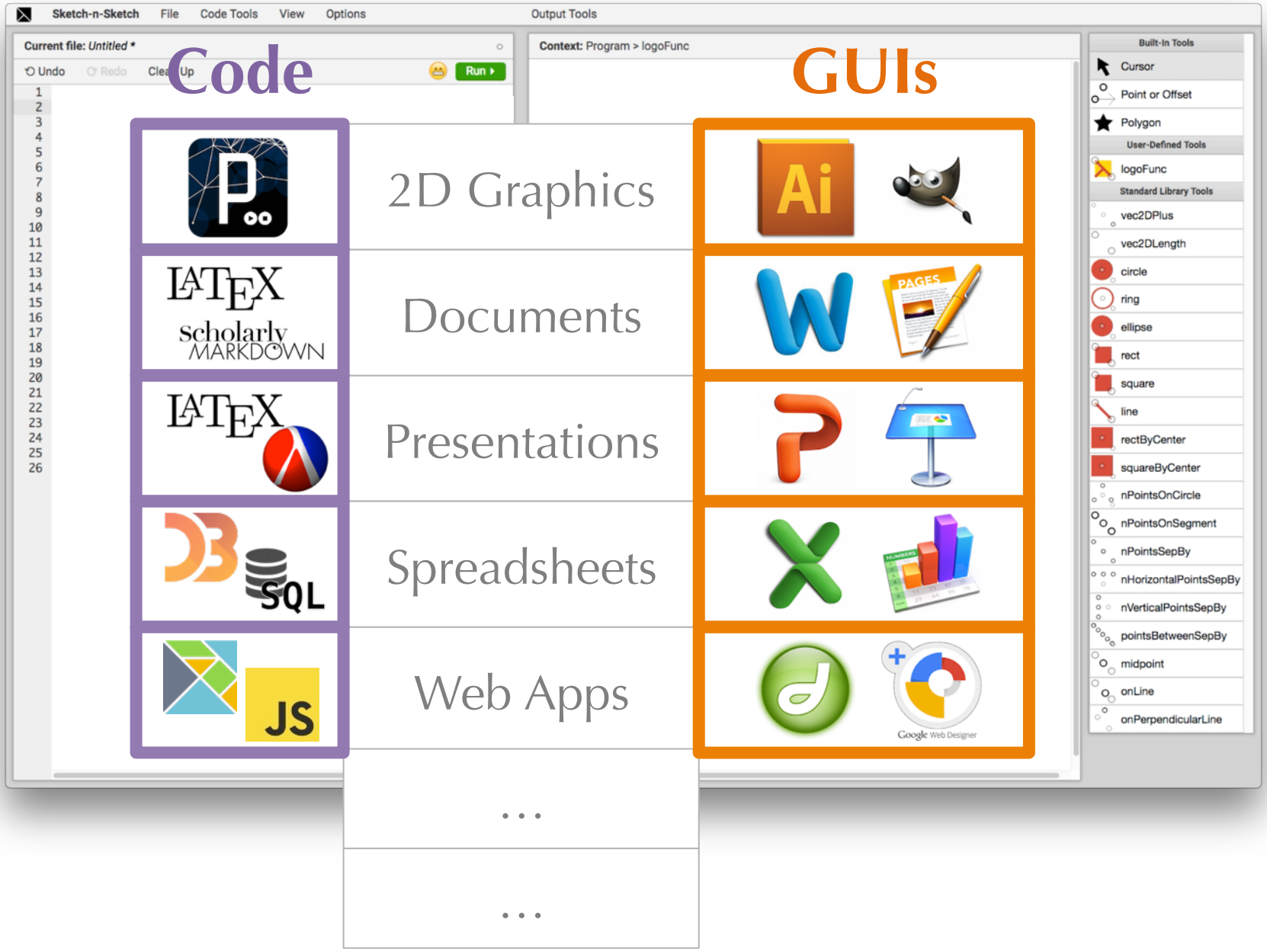
PL + UI

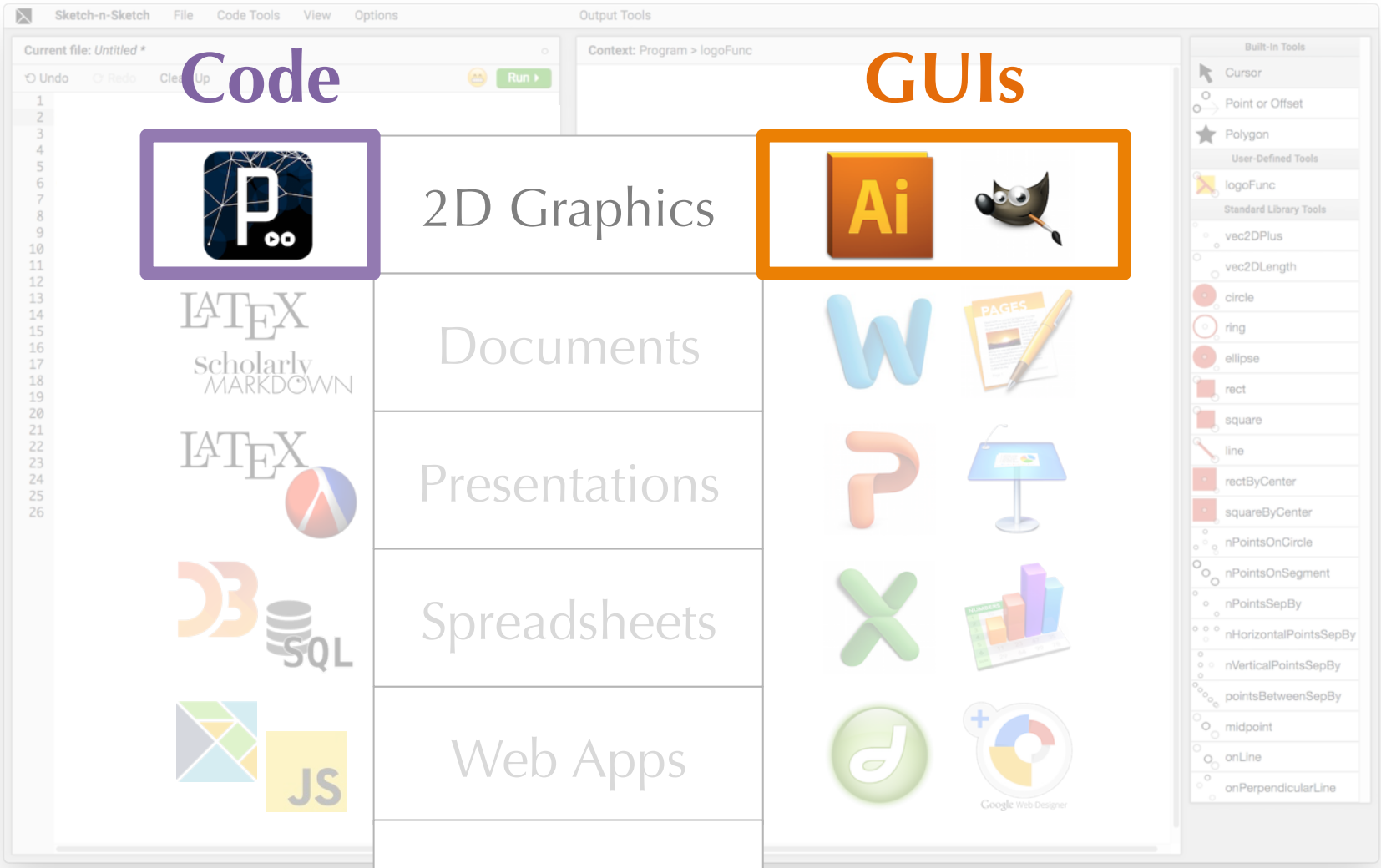
Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]


Code
GUIs
2D Graphics
Documents
Presentations
Spreadsheets
Web Apps
...
...
Sketch-n-Sketch
File
Code Tools
View
Options
Output Tools
Current file: Untitled *
Undo
Redo
Clean Up
Run
Context: Program > logoFunc
Built-In Tools
Cursor
Point or Offset
Polygon
User-Defined Tools
logoFunc
Standard Library Tools
vec2DPlus
vec2DLength
circle
ring
ellipse
rect
square
line
rectByCenter
squareByCenter
nPointsOnCircle
nPointsOnSegment
nPointsSepBy
nHorizontalPointsSepBy
nVerticalPointsSepBy
pointsBetweenSepBy
midpoint
onLine
onPerpendicularLine


Code
GUIs
2D Graphics
Documents
Presentations
Spreadsheets
Web Apps
...
...
Sketch-n-Sketch
File
Code Tools
View
Options
Output Tools
Current file: Untitled *
Undo
Redo
Clean Up
Run
Context: Program > logoFunc
Built-In Tools
Cursor
Point or Offset
Polygon
User-Defined Tools
logoFunc
Standard Library Tools
vec2DPlus
vec2DLength
circle
ring
ellipse
rect
square
line
rectByCenter
squareByCenter
nPointsOnCircle
nPointsOnSegment
nPointsSepBy
nHorizontalPointsSepBy
nVerticalPointsSepBy
pointsBetweenSepBy
midpoint
onLine
onPerpendicularLine

Creative Coding

p5.js

p5* File Edit Sketch Help Log in or Sign up

Auto-refresh maquina by mecobi

sketch.js

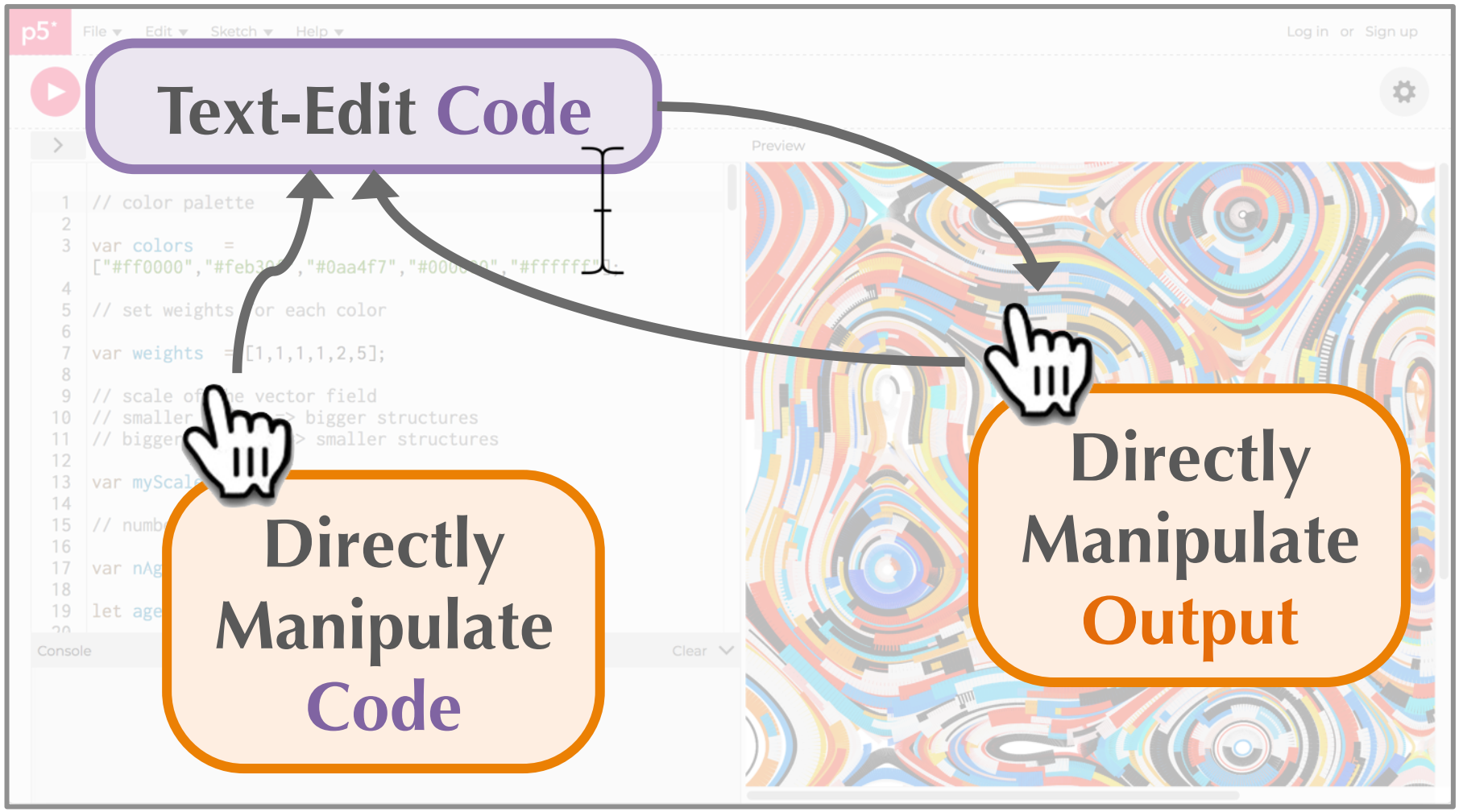
```
// color palette

var colors   =
["#ff0000","#feb30f","#0aa4f7","#000000","#ffffff"];

// set weights for each color

var weights  = [1,1,1,1,2,5];

// scale of the vector field
// smaller values => bigger structures
// bigger values  ==> smaller structures

var myScale =2;

// number of drawing agents

var nAgents = 2000;

let agent = [];
```

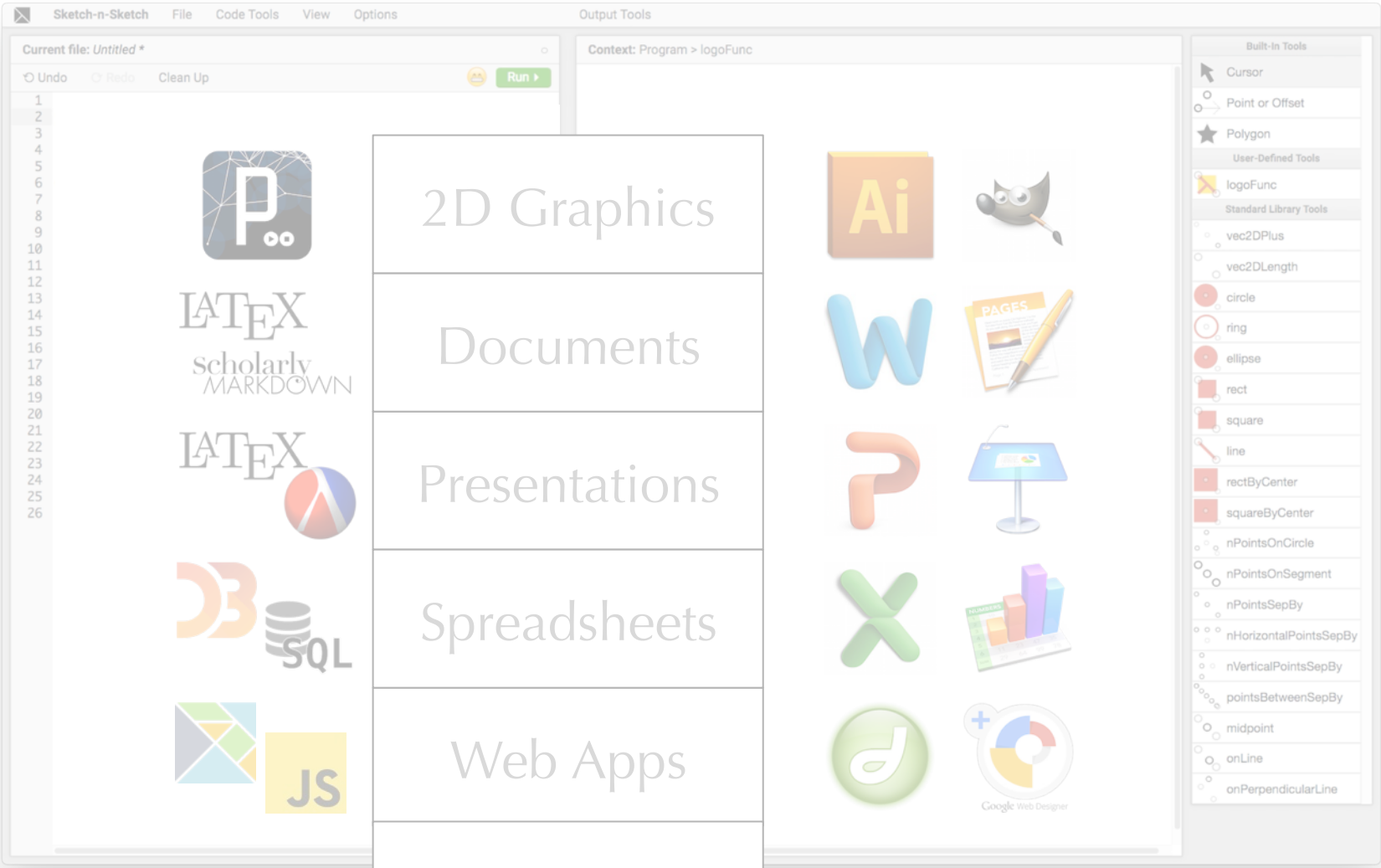
Console Clear

Preview

Creative Coding

p5.js





2D Graphics

Documents

Presentations

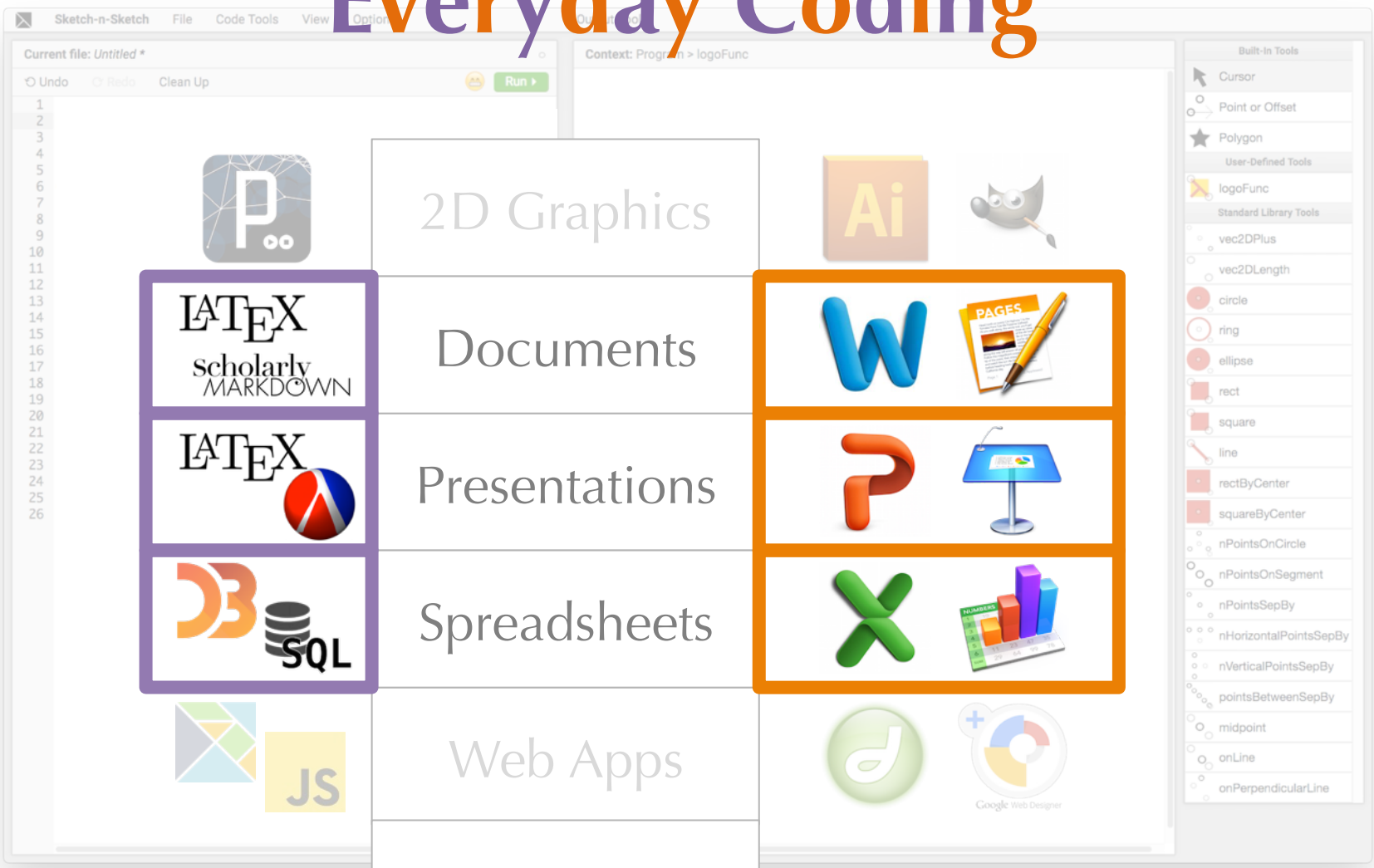
Spreadsheets

Web Apps

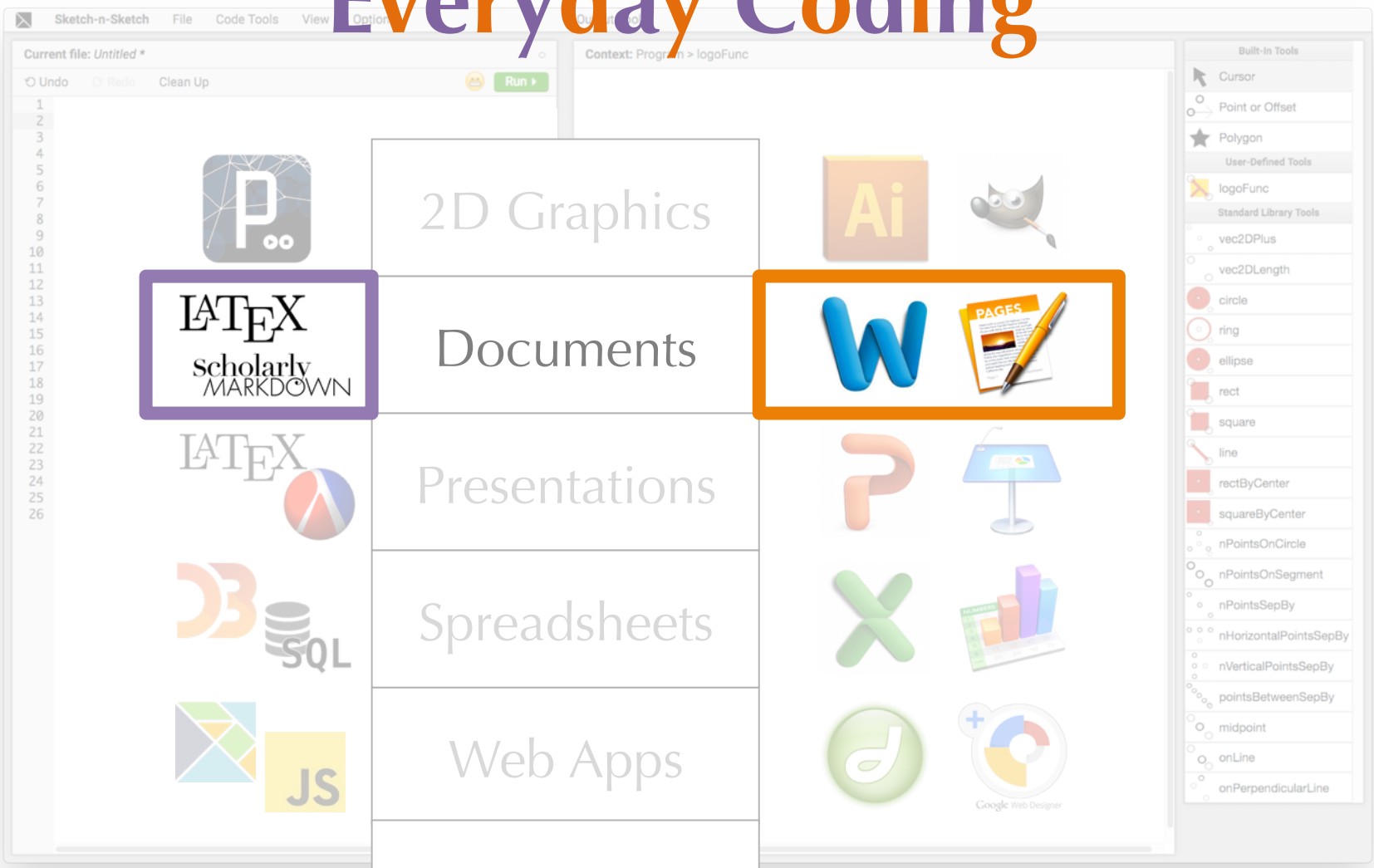
...

...

"Everyday Coding"



"Everyday Coding"



2D Graphics

Documents

Presentations

Spreadsheets

Web Apps

...

...

"Everyday Coding"

Mikaël Mayer

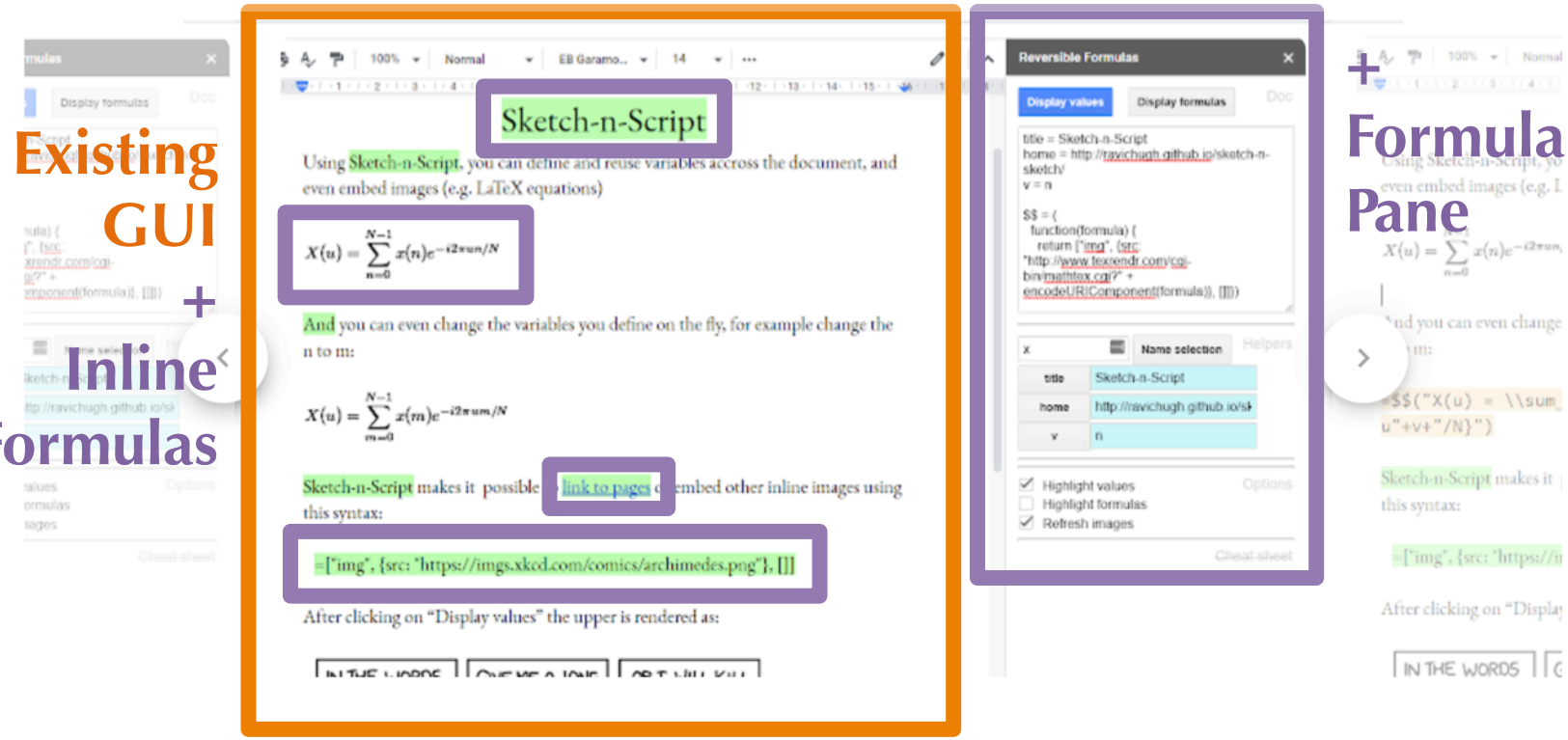
=X Spreadsheet Cells i...

MikaelMayer

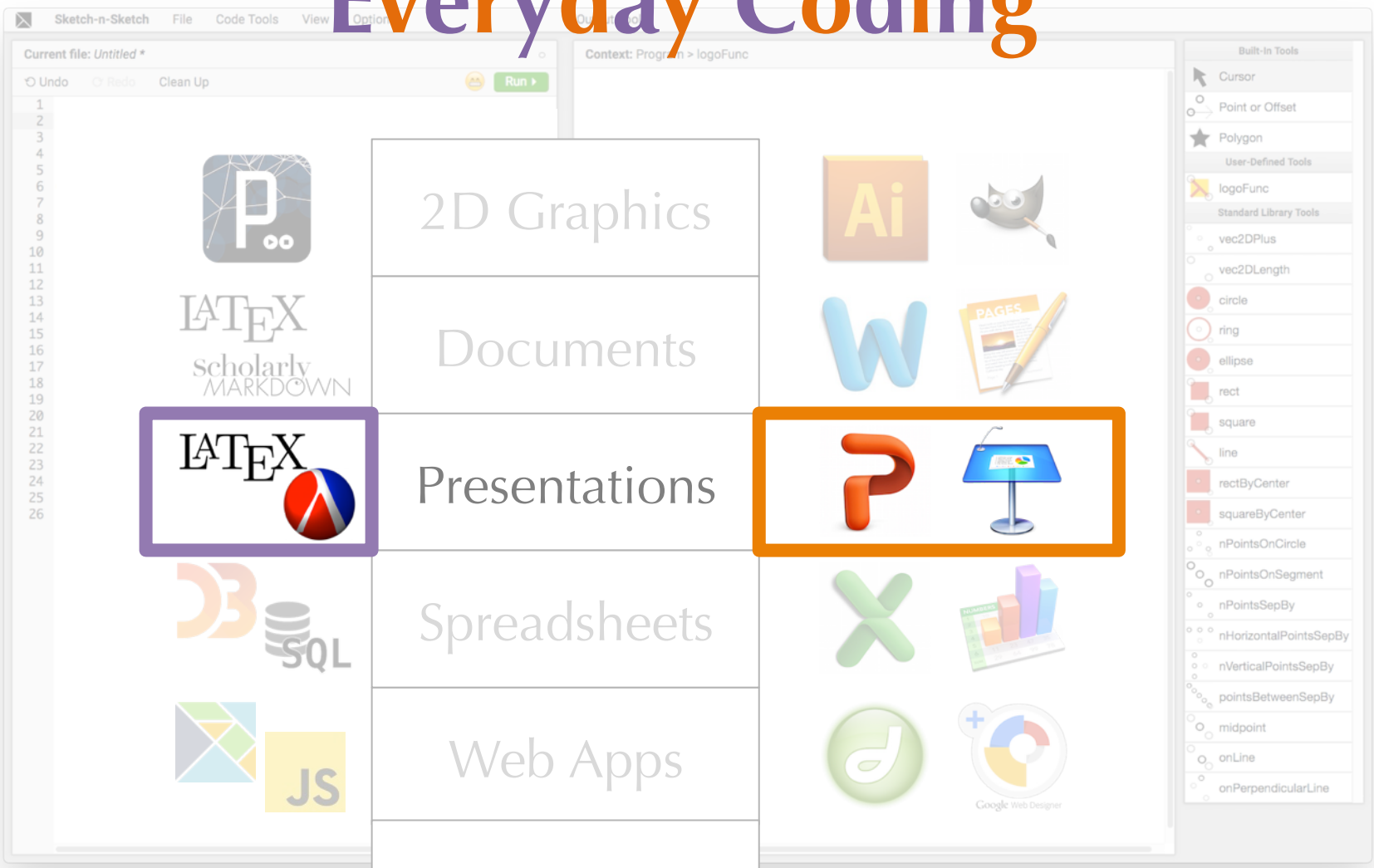
Docs Add-on

★★★½☆ (21) · 4,880

Install



"Everyday Coding"



2D Graphics

Documents

Presentations

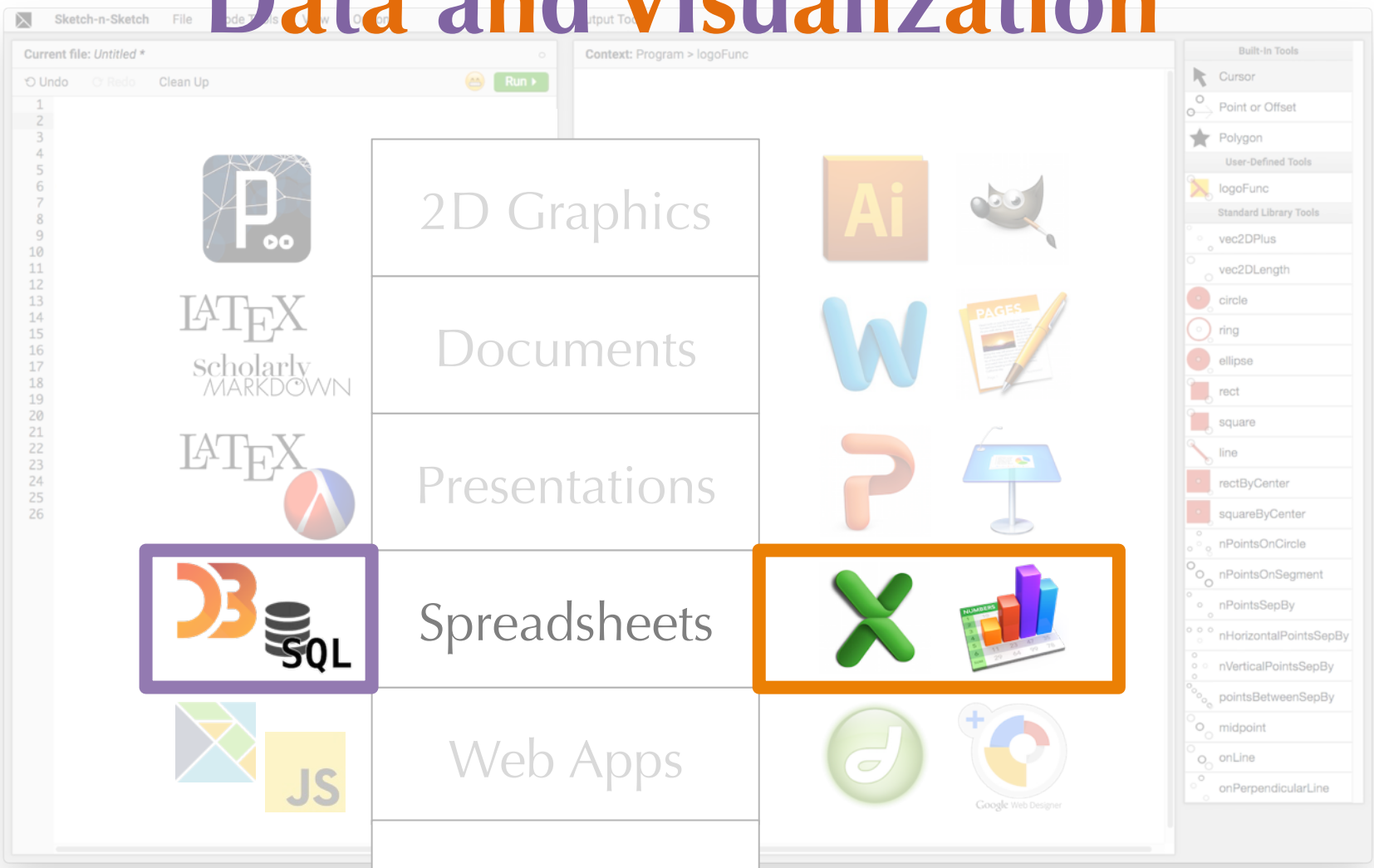
Spreadsheets

Web Apps

...

...

Data and Visualization



2D Graphics

Documents

Presentations

Spreadsheets

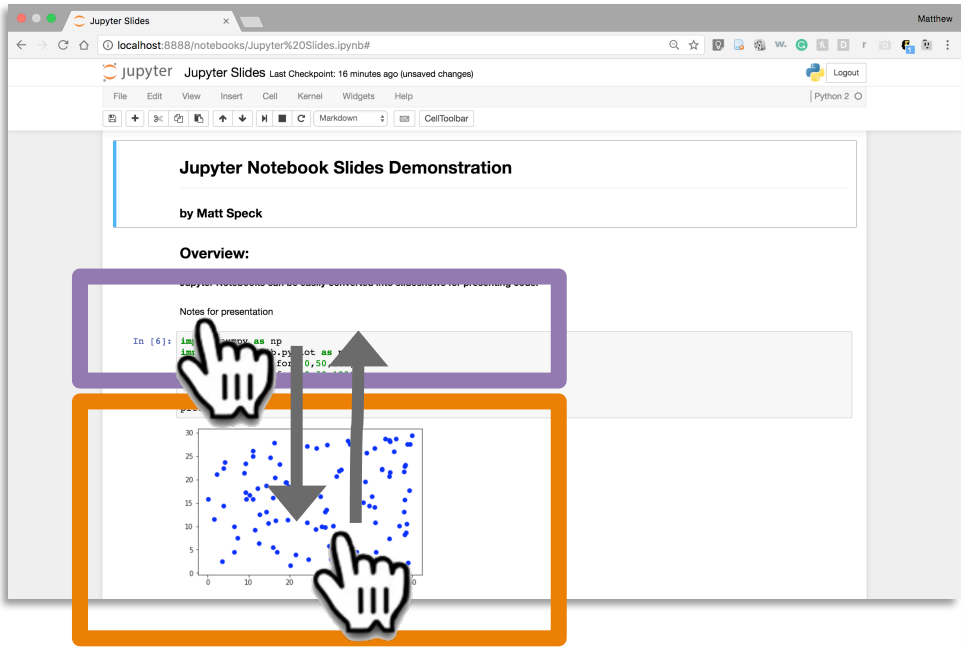
Web Apps

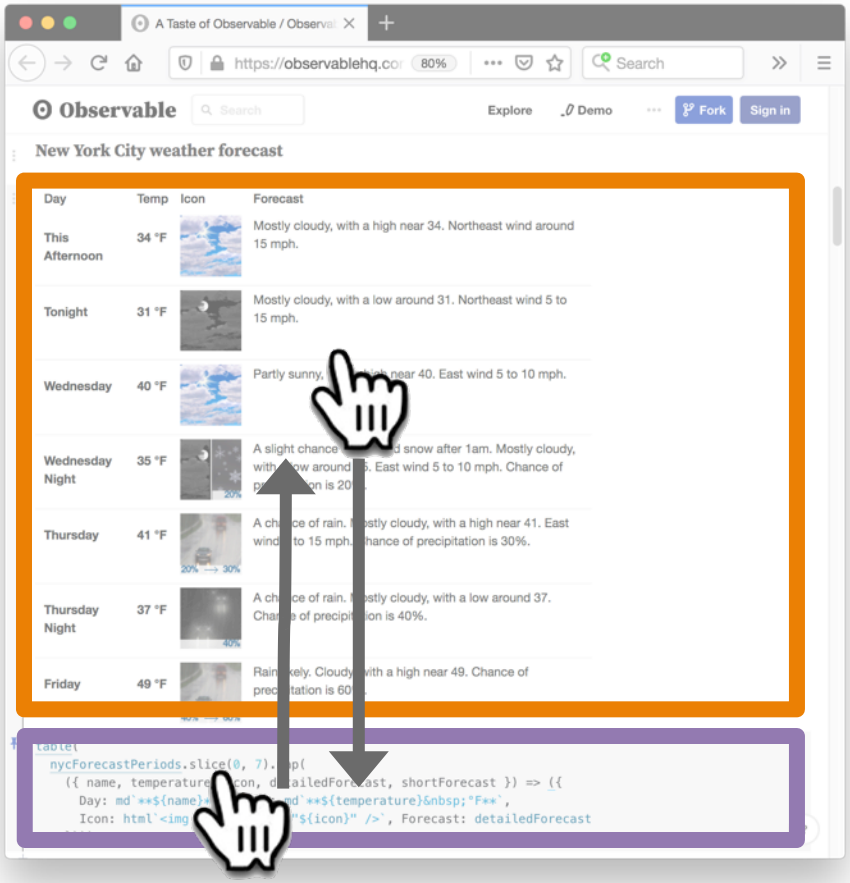
...

...

Data and Visualization

Computational Notebooks





[>>> Skip Ivy to Spreadsheets >>>]

Data and Visualization

Ivy: Multi-Modal Vis. Editor

[CHI 2021]

Andrew McNutt

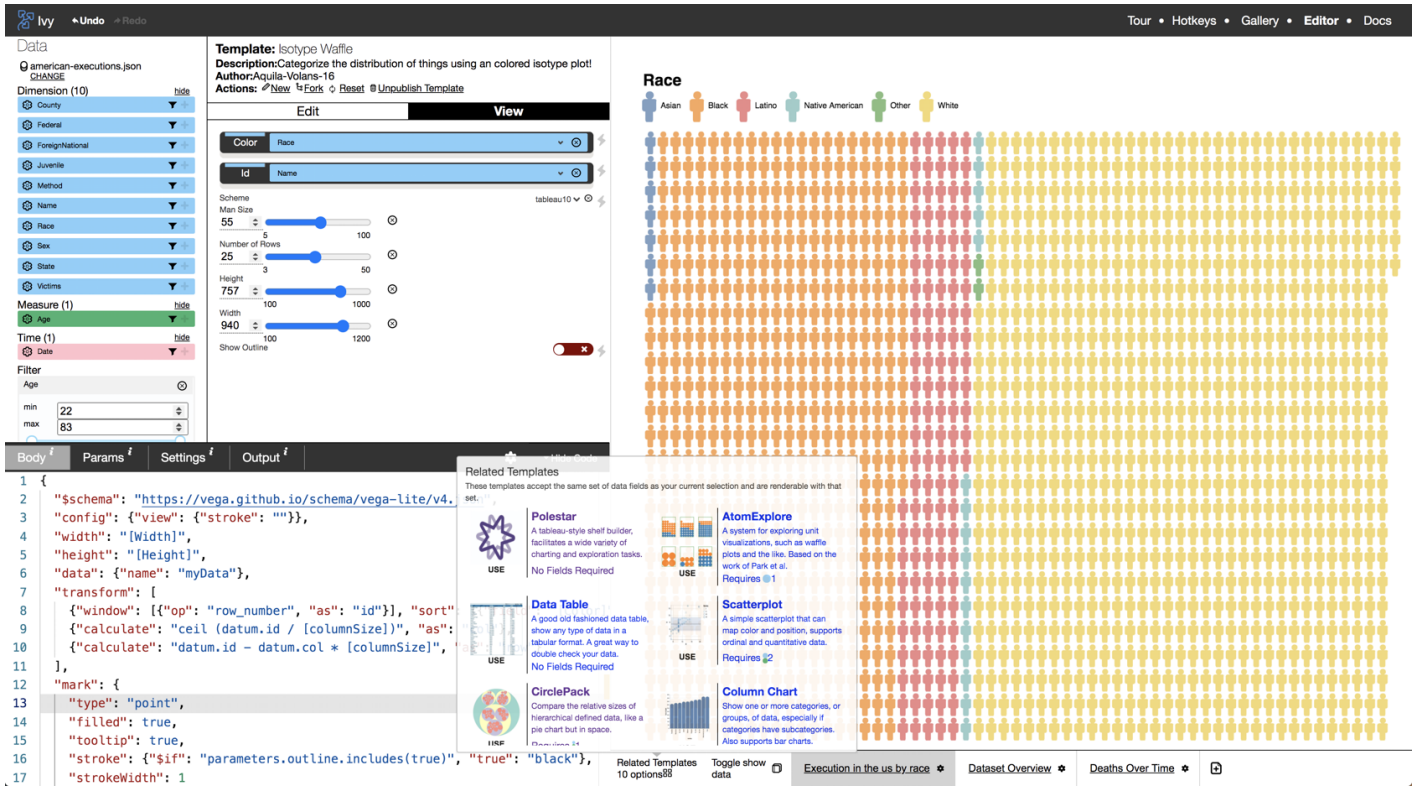
	Ease of Use	Explorability	Flexibility	Ease of Reuse
Chart choosers	🙂	☹️	☹️	n/a
Shelf builders	😐	🤩	😐	☹️
Text (declarative)	☹️	☹️	🙂	😐
Text (procedural)	😫	😫	🤩	🤩

Data and Visualization

Ivy: Multi-Modal Vis. Editor

[CHI 2021]

Andrew McNutt

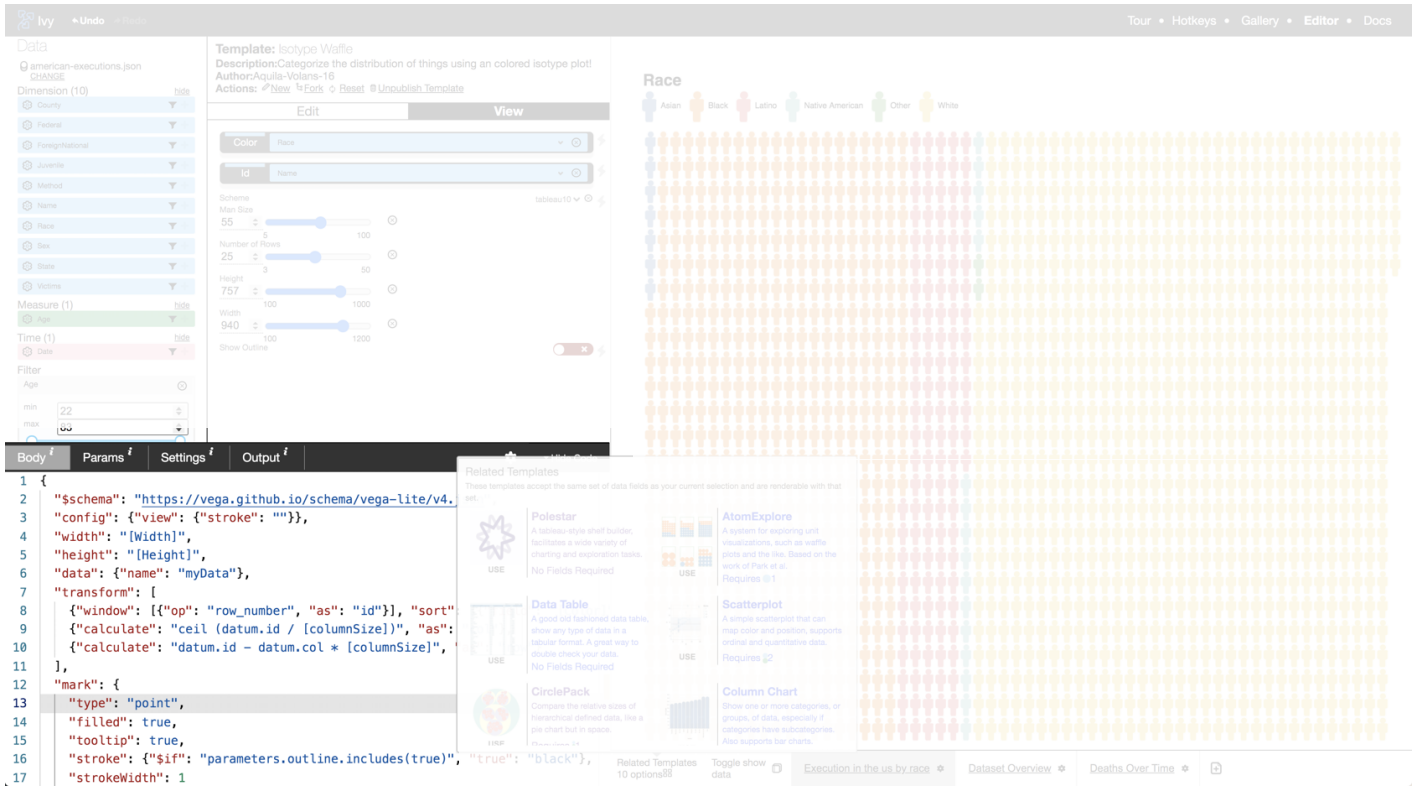


Data and Visualization

Ivy: Multi-Modal Vis. Editor

[CHI 2021]

Andrew McNutt



Template = abstracted Vega/Vega-Lite definition

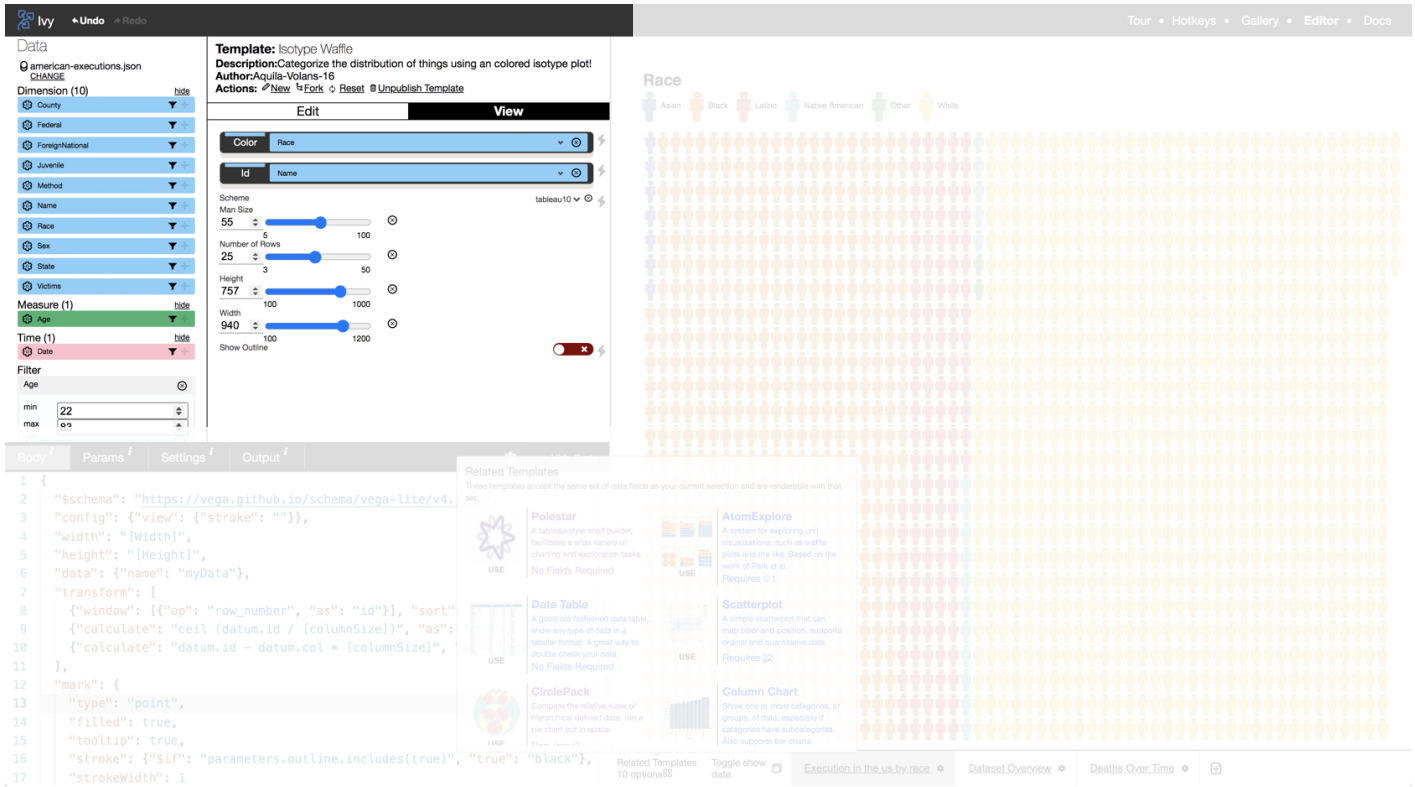
Data and Visualization

Ivy: Multi-Modal Vis. Editor

[CHI 2021]

Andrew McNutt

Shelf-builder UI for template arguments



Template = abstracted Vega/Vega-Lite definition

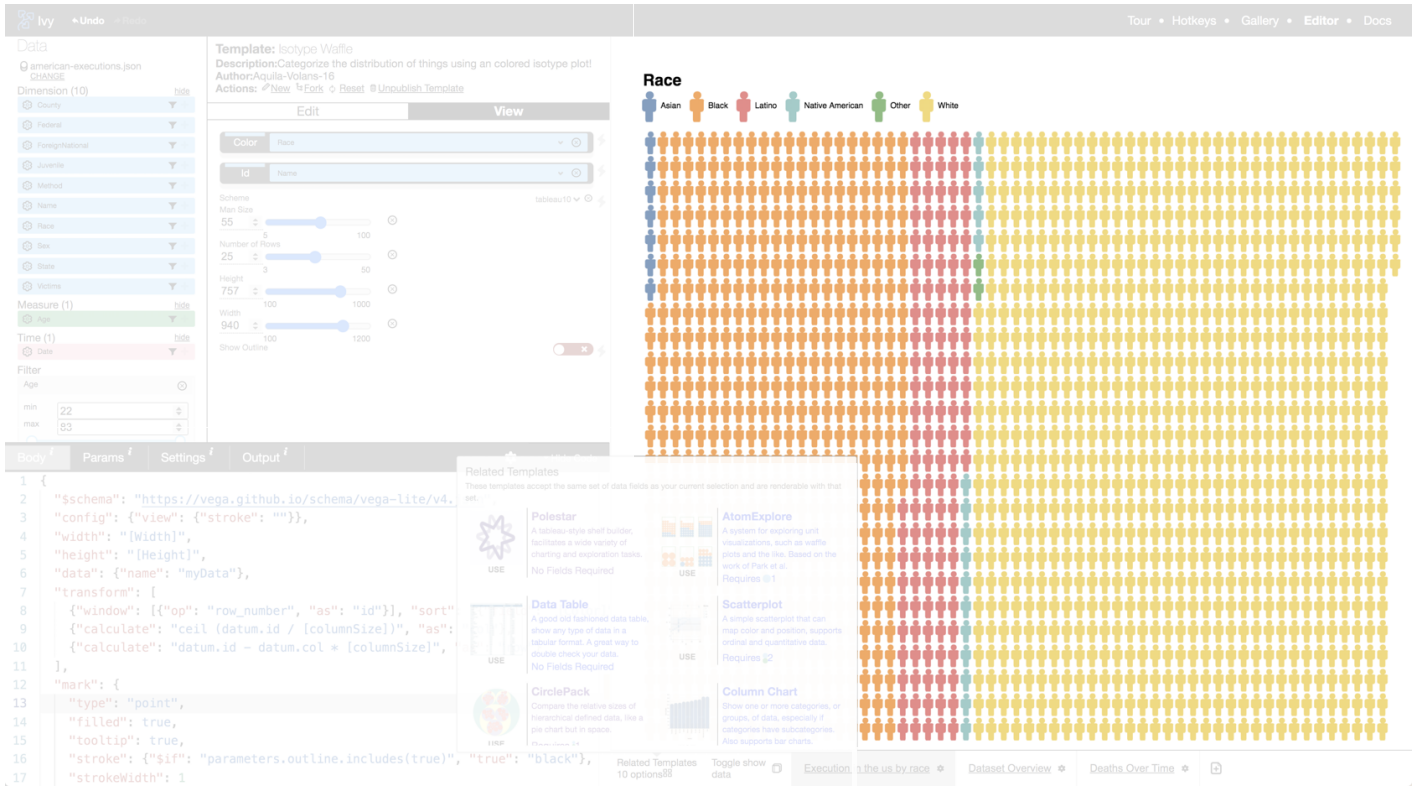
Data and Visualization

Ivy: Multi-Modal Vis. Editor

[CHI 2021]

Andrew McNutt

Shelf-builder UI for template arguments



Template = abstracted Vega/Vega-Lite definition

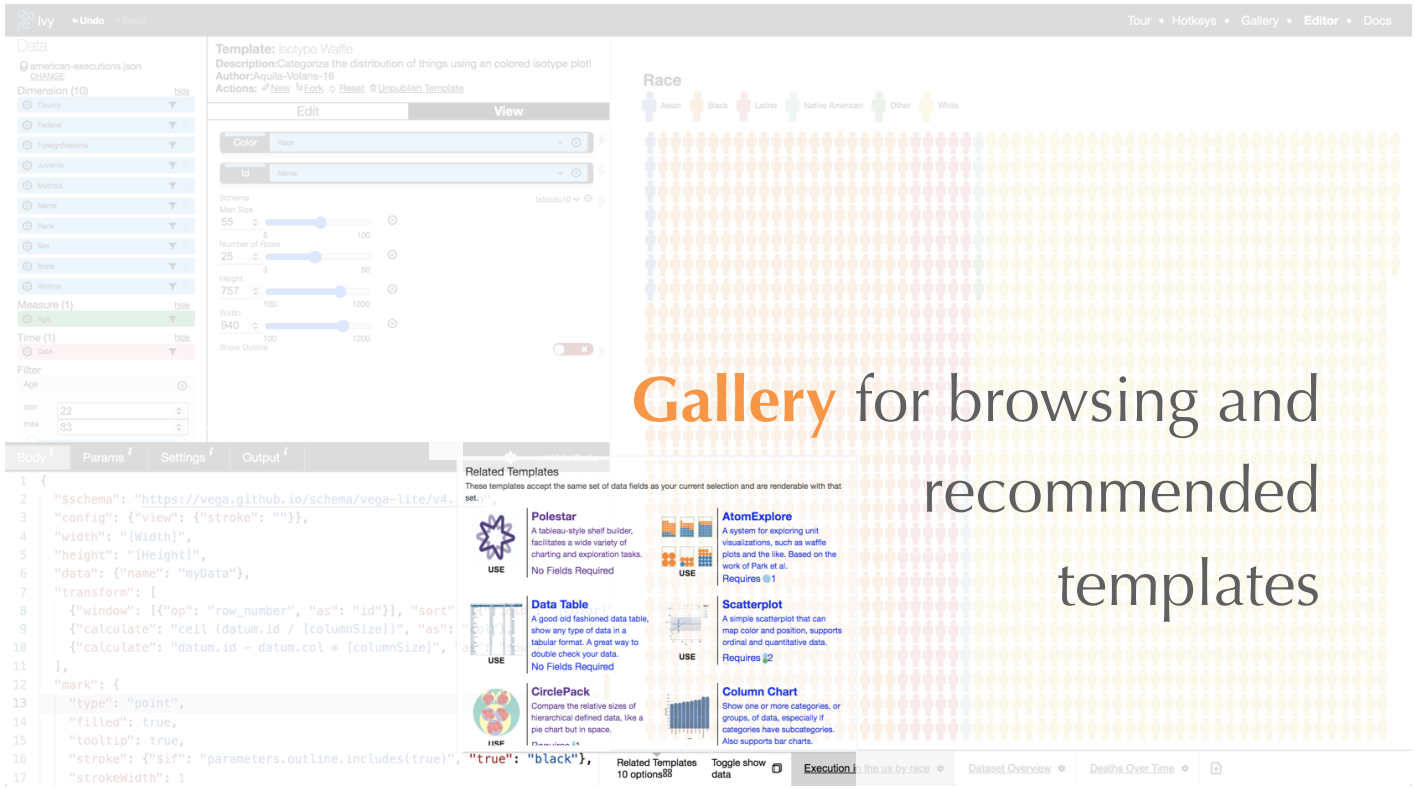
Data and Visualization

Ivy: Multi-Modal Vis. Editor

[CHI 2021]

Andrew McNutt

Shelf-builder UI for template arguments



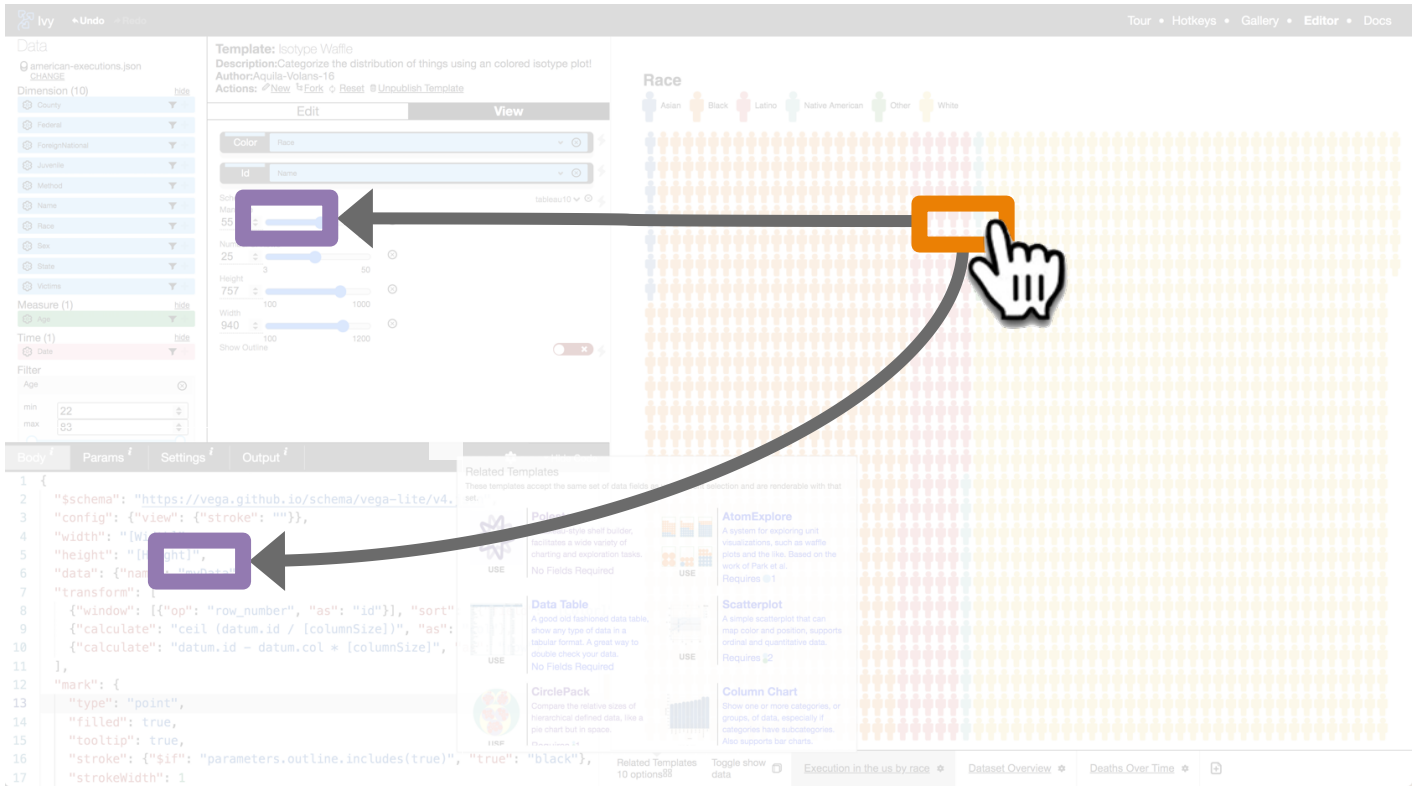
Gallery for browsing and recommended templates

Template = abstracted Vega/Vega-Lite definition

Data and Visualization

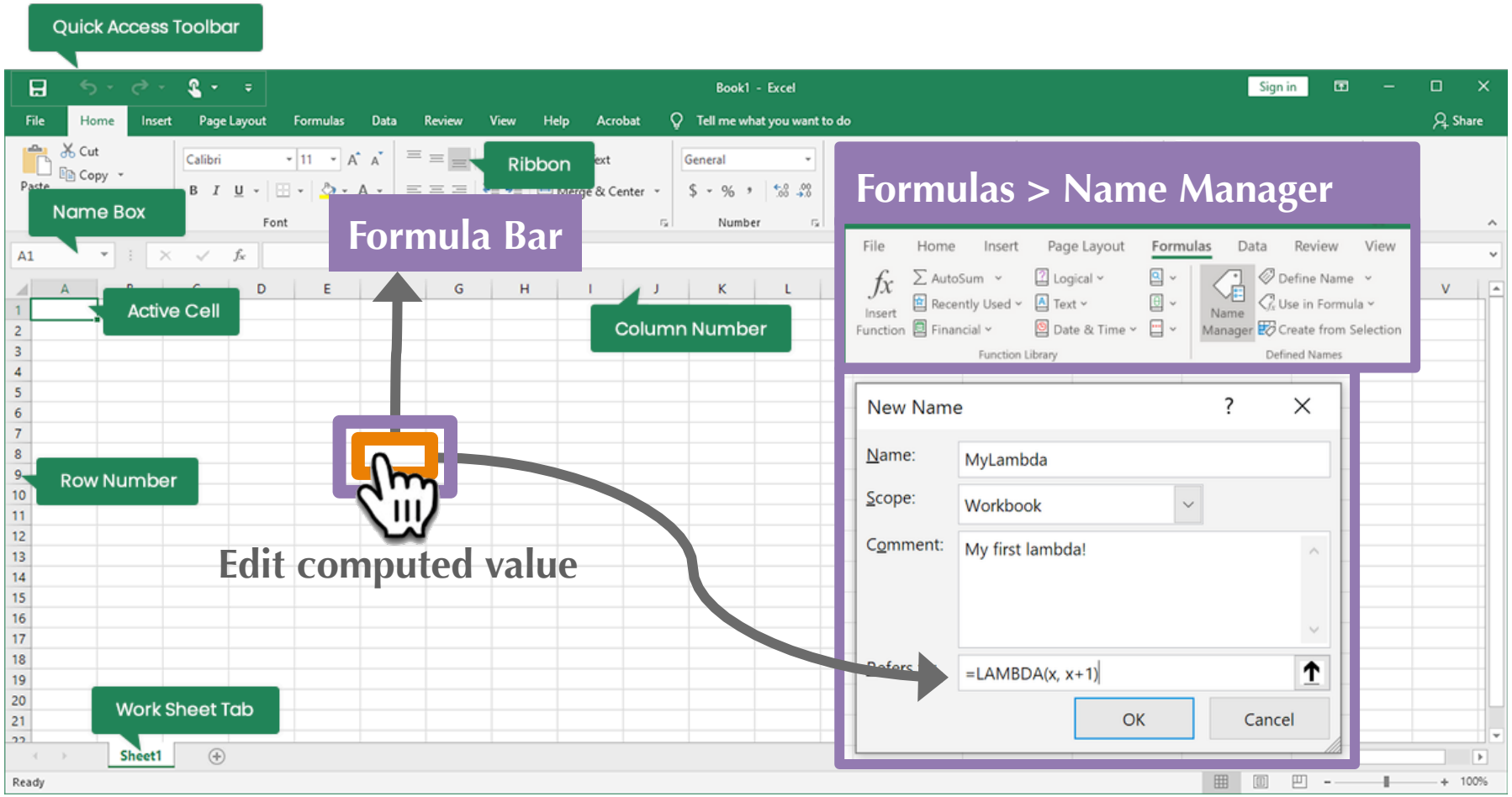
Ivy: Multi-Modal Vis. Editor

Andrew McNutt

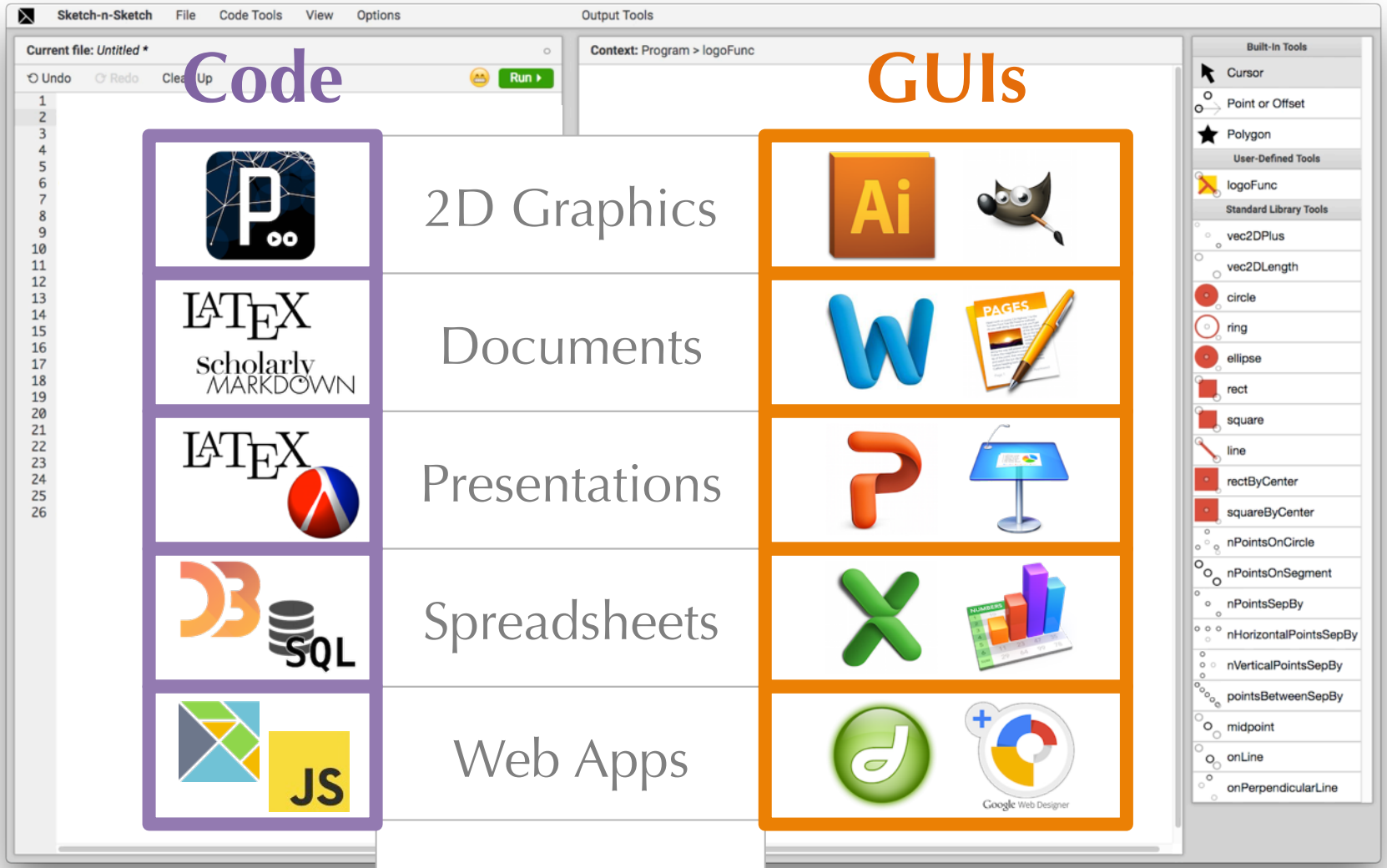


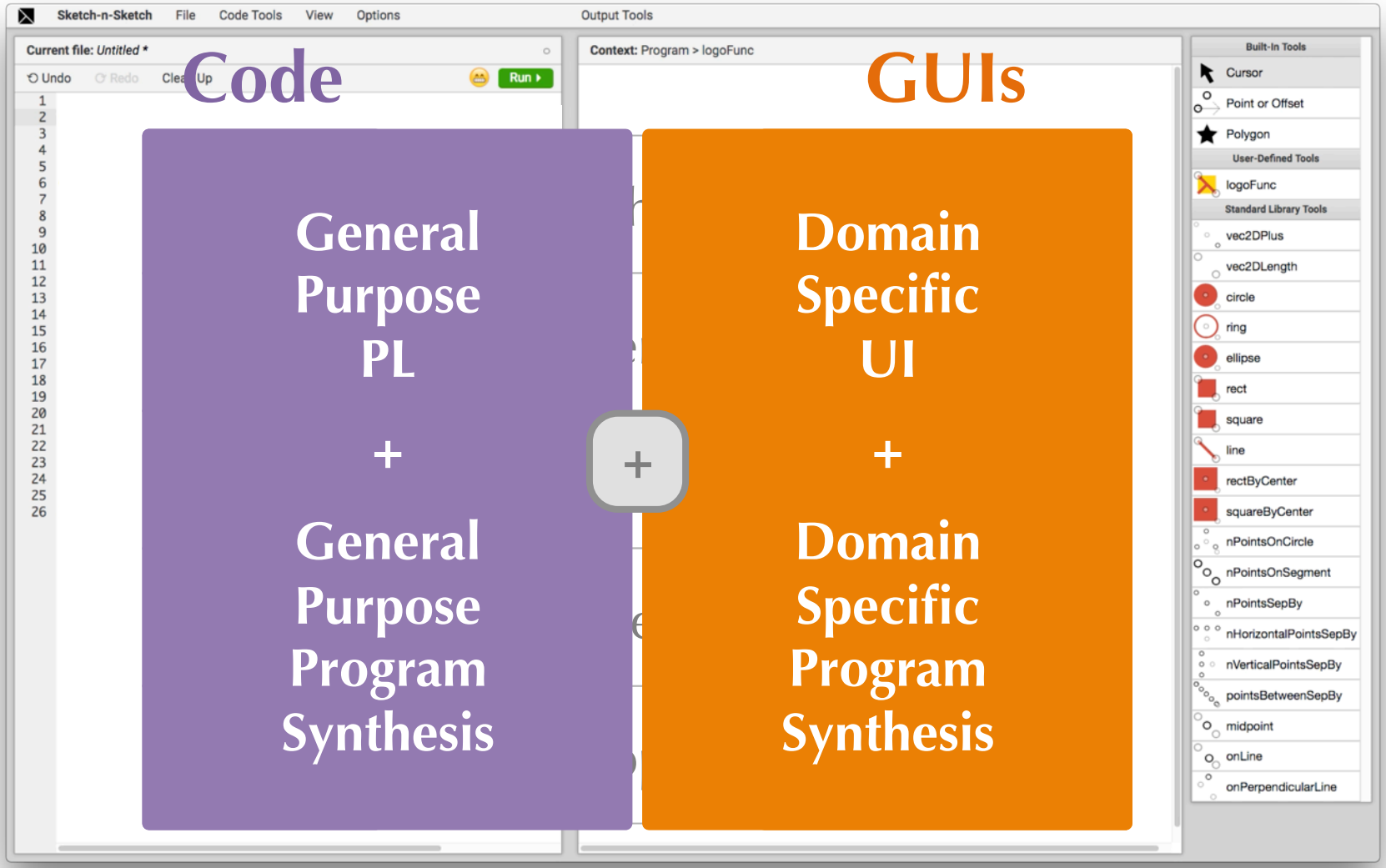
Data and Visualization

Spreadsheets



https://www.trainingconnection.com/excel/lessons/excel-interface.php
https://techcommunity.microsoft.com/t5/excel-blog/announcing-lambda-turn-excel-formulas-into-custom-functions/ba-p/1925546


Sketch-n-Sketch
File
Code Tools
View
Options
Output Tools
Current file: Untitled *
Undo
Redo
Clear Up
Run
Context: Program > logoFunc
Code
GUIs
Built-In Tools
Cursor
Point or Offset
Polygon
User-Defined Tools
logoFunc
Standard Library Tools
vec2DPlus
vec2DLength
circle
ring
ellipse
rect
square
line
rectByCenter
squareByCenter
nPointsOnCircle
nPointsOnSegment
nPointsSepBy
nHorizontalPointsSepBy
nVerticalPointsSepBy
pointsBetweenSepBy
midpoint
onLine
onPerpendicularLine
LaTeX
Scholarly MARKDOWN
LaTeX
SQL
JS
Ai
PAGES
Google Web Designer
2D Graphics
Documents
Presentations
Spreadsheets
Web Apps
...
...
Text
Trees/Blocks



Text

Trees/Blocks

Thanks!

Nick Collins

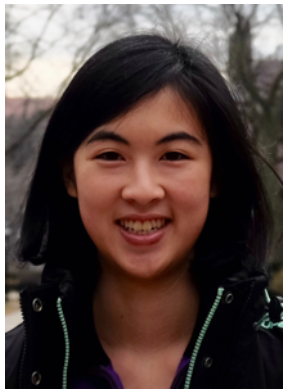
Brian Hempel

Justin Lubin

Mikaël Mayer

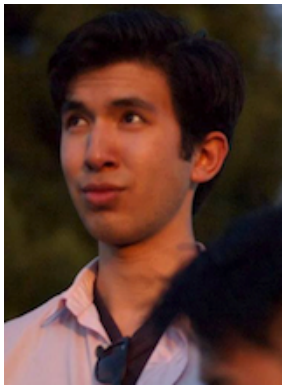
Cyrus Omar

Andrew McNutt



Grace Lu

Mitch Spradlin



Jacob Albers

Swiss National Science Foundation

Bidirectional Evaluation

[PLDI 2016, OOPSLA 2018]

Live Programming by Example

[POPL 2019, ICFP 2020]

Sketch-n-Sketch

http://ravichugh.github.io/sketch-n-sketch/

Hybrid Text-GUI Code Editors

[ICSE 2018, VL/HCC 2020]

Output-Directed Programming

[UIST 2016, UIST 2019]

EXTRA SLIDES

Live Programming by Example

```
length : List a -> Int
length =
  ??

assert (length [2,0,1,6] == 4)
assert (length       [] == 0)
```

Program Synthesizer

```
length xs =
  case xs of
    []     -> 0
    x::xs' -> 1 + length xs'
```

```
stutter : List a -> List a
stutter =
  ??

assert (stutter [1,2,3] == [1,1,2,2,3,3])
assert (stutter      [] == []           )
```

Program Synthesizer

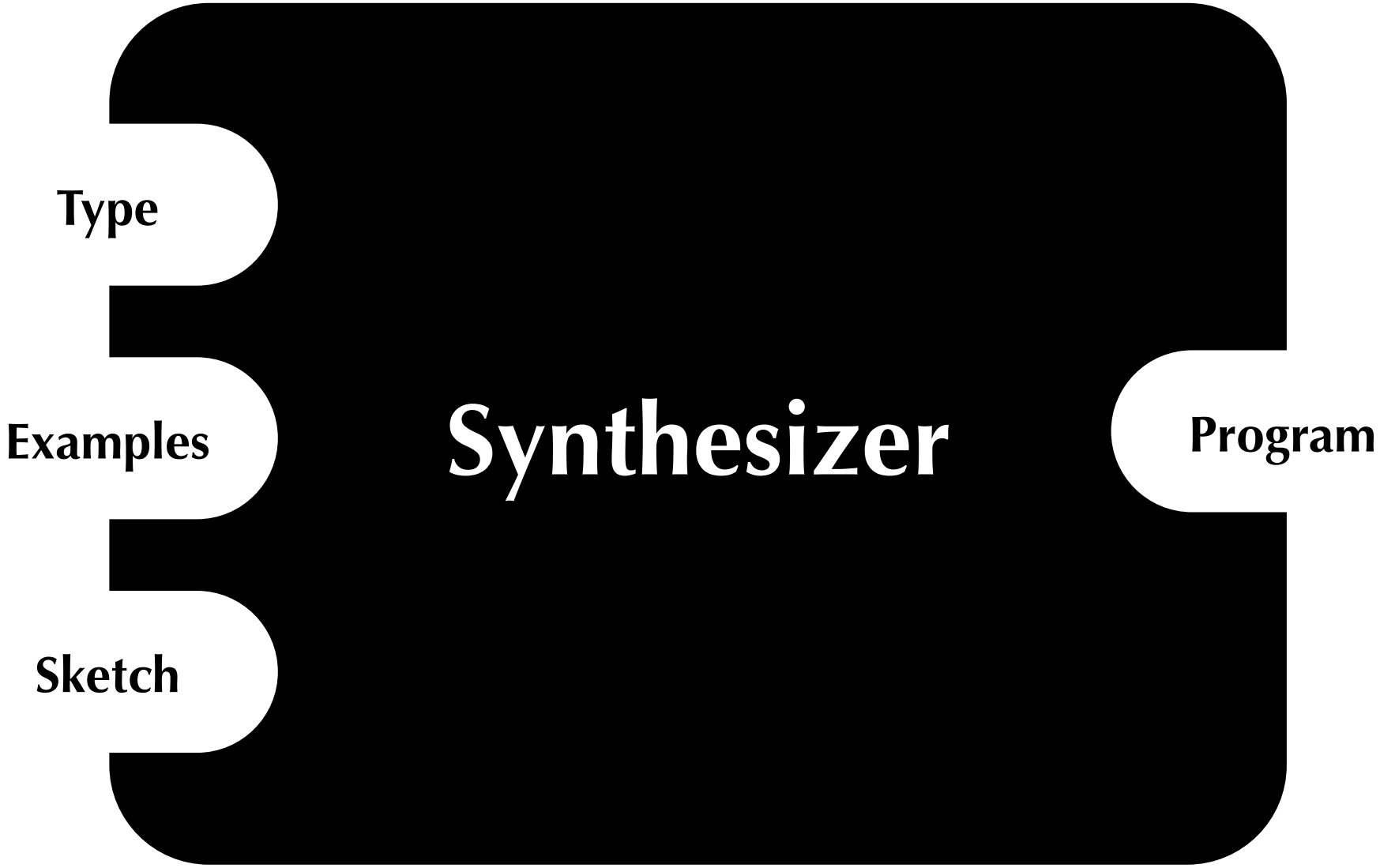
```
stutter xs =
  case xs of
    []     -> []
    x::xs' -> x :: x :: stutter xs'
```

```
max : Nat -> Nat -> Nat
max n m =
  case n of
    Zero    -> m
    Succ n' -> ??

assert (max 1 1 == 1)
assert (max 1 2 == 2)
assert (max 2 0 == 2)
assert (max 2 1 == 2)
assert (max 2 2 == 2)
```

Program Synthesizer

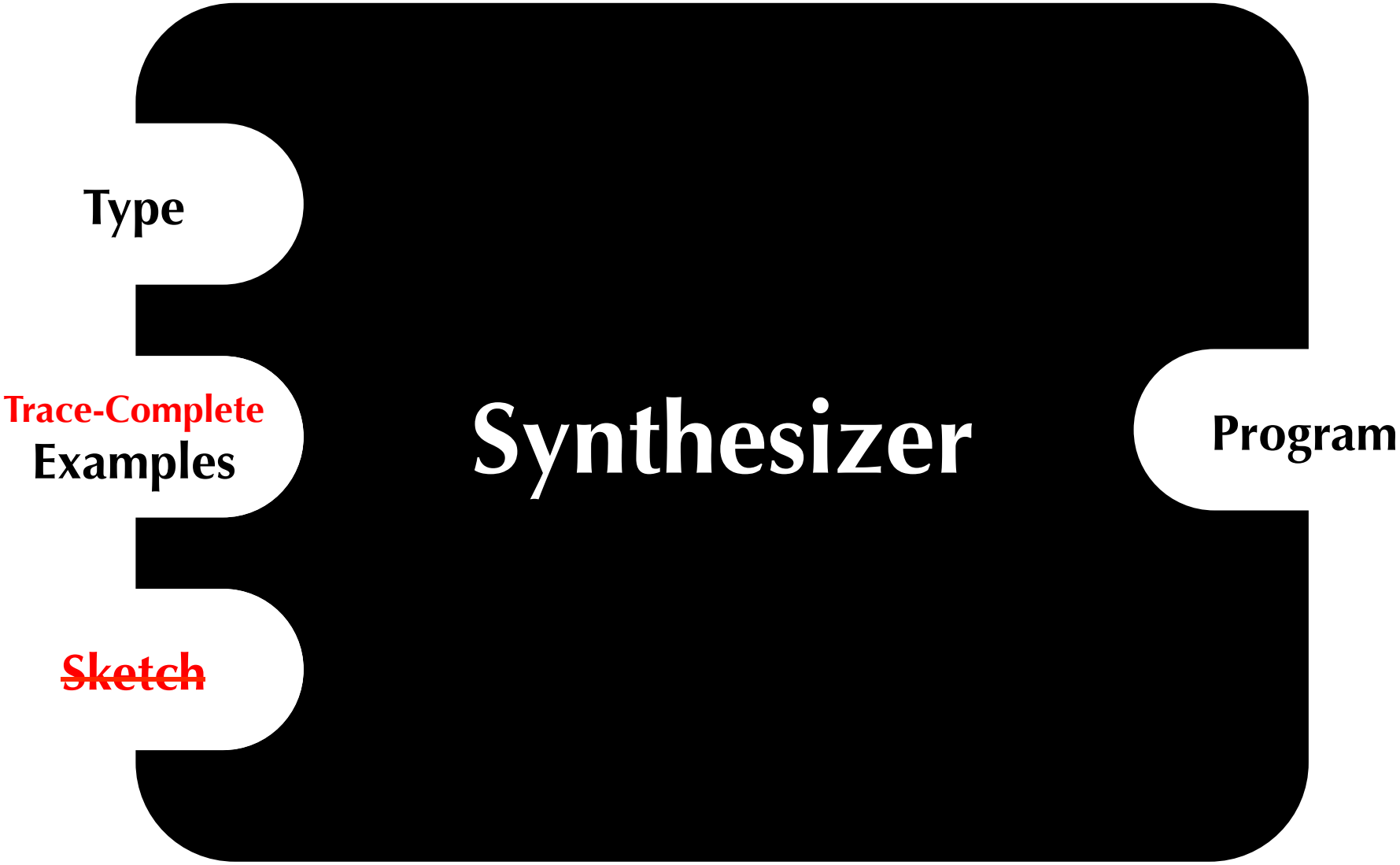
↓

```
max n m =
  case n of
    Zero    -> m
    Succ n' -> case m of
                 Zero    -> n
                 Succ m' -> Succ (max n' m')
```


Type
Examples
Sketch
Synthesizer
Program

Sketch, Rosette, … Leon, Synquid, … FlashFill, PROSE, …

Myth [Osera et al. PLDI 2015, POPL 2016]



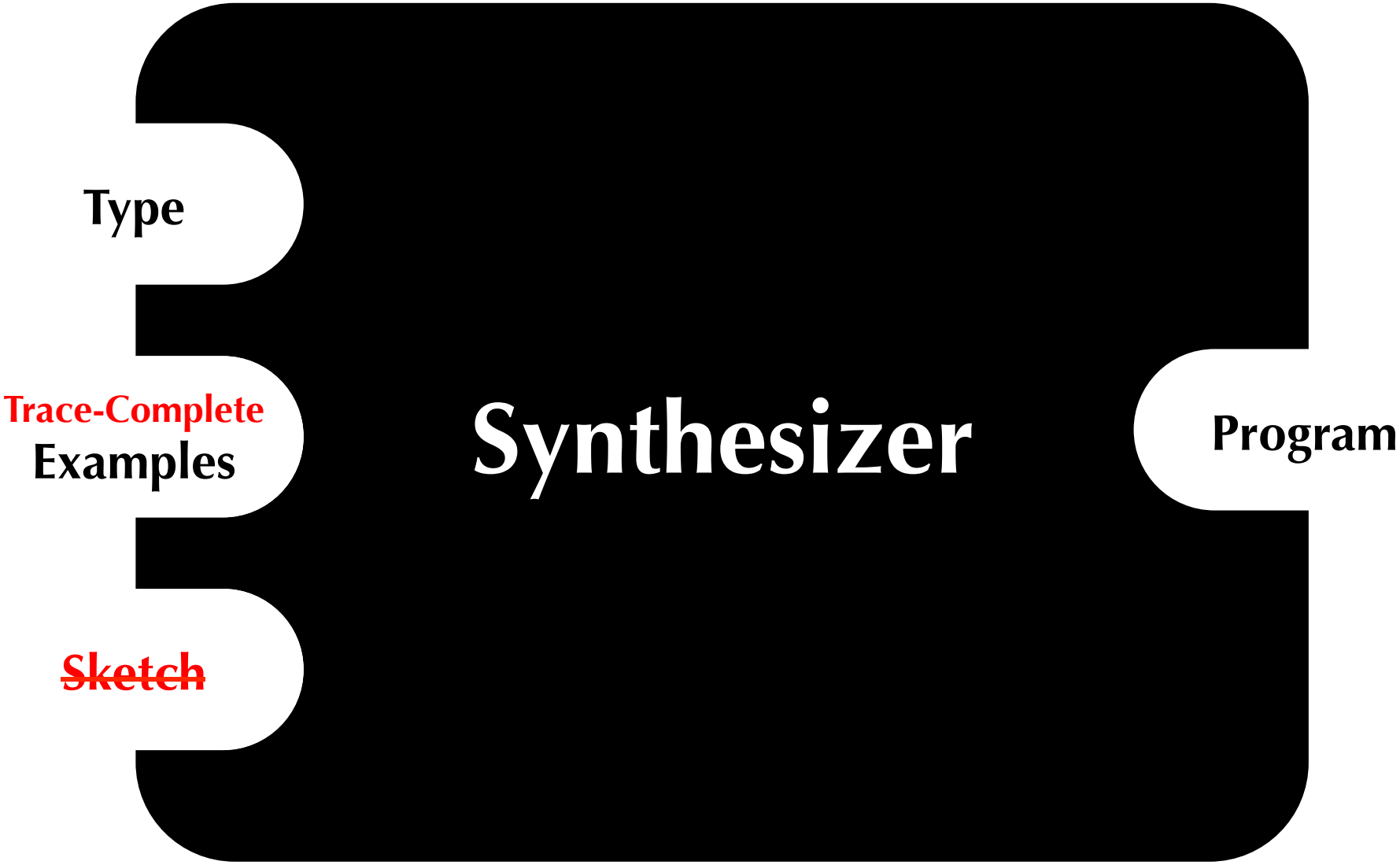
Trace-Complete
Examples

```
length : List a -> Int
length =
  ??

assert (length [2,0,1,6] == 4)
assert (length        [] == 0)
```

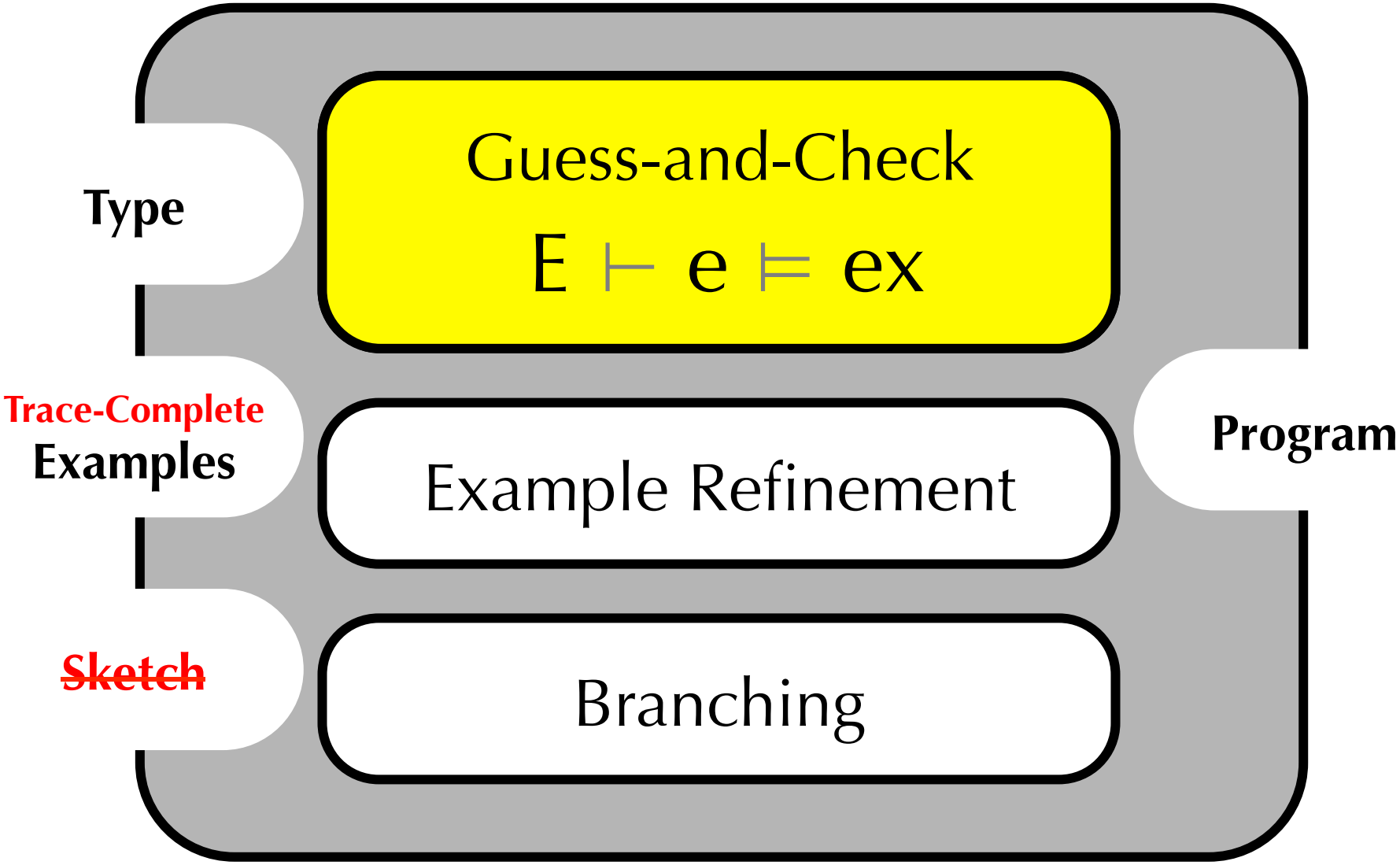
Trace-Complete
Examples

```
length : List a -> Int
length =
  ??

assert (length [2,0,1,6] == 4)
assert (length   [0,1,6] == 3)
assert (length     [1,6] == 2)
assert (length       [6] == 1)
assert (length        [] == 0)
```

Myth [Osera et al. PLDI 2015, POPL 2016]



Myth [Osera et al. PLDI 2015, POPL 2016]



Example Satisfaction in Myth

$$E \vdash e \models ex$$

Eval		**Check**
$E \vdash e \Rightarrow v$	+	$v \models ex$

$$\begin{aligned} v &::= \ldots \mid \lambda x.e \\ ex &::= \ldots \mid \{ v_{in} \rightarrow ex_{out} \} \end{aligned}$$

Input-output examples for functions

$$E \vdash e \models ex$$

How to **Eval** and **Check** Sketches ??

Eval

$$E \vdash e \Rightarrow v$$

$+$

Check

$$v \models ex$$

$$E \vdash e \vDash ex$$

How to **Eval** and **Check** Sketches ??

Live Eval

$$E \vdash e \Rightarrow r$$

+

Check

$$v \vDash ex$$

Partially evaluate sketch to result r (value or "paused")

$$E \vdash e \models ex$$

How to **Eval** and **Check** Sketches ??

Live Eval

$$E \vdash e \Rightarrow r$$

+

Live Uneval

$$r \Leftarrow ex \dashv X$$

Partially evaluate sketch to result r (value or "paused")

Constraints X s.t. r will resume and satisfy ex

$$E \vdash e \Rightarrow r \Leftarrow ex \dashv X$$

Live Bidirectional Evaluation

Live Eval

$$E \vdash e \Rightarrow r$$

+

Live Uneval

$$r \Leftarrow ex \dashv X$$

Partially evaluate sketch to result r (value or "paused")

Constraints X s.t. r will resume and satisfy ex

>>> Skip Examples >>>

Live Eval

$$E \vdash e \Rightarrow r$$

Partially evaluate sketch to result r (value or "paused")

Eval

$e \Rightarrow v$

Expressions

Values **Errors**

2019 + 1 ⇒ **2020**

2019 ++ "!" ⇒ **crash**

(??, 2019 + 1) ⇒ **crash**

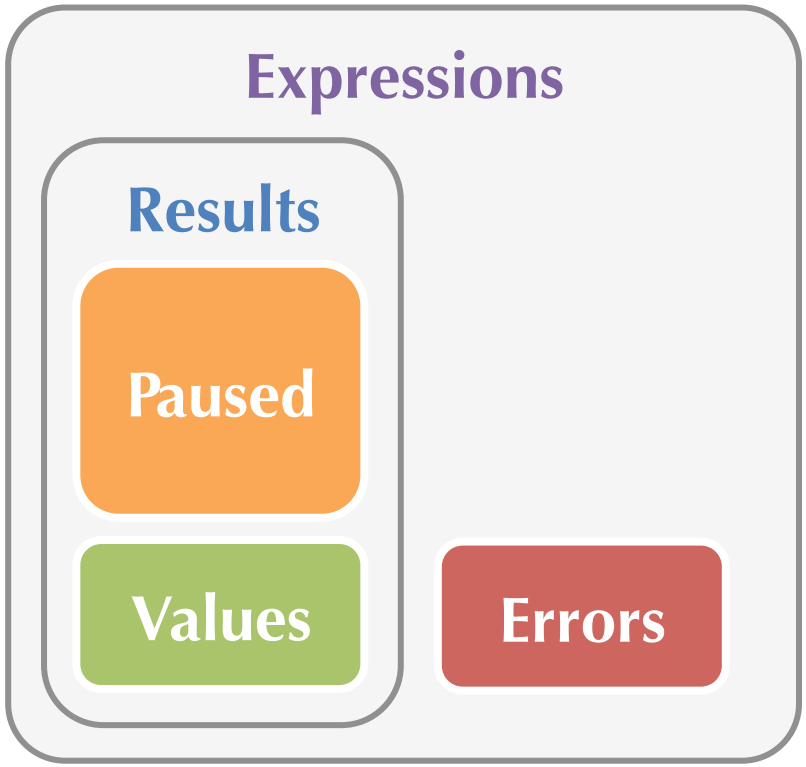
Holes typically represented as

`raise "Not yet implemented"`

Live Eval

$$e \Rightarrow r$$

Partially evaluate "around" holes



(??, 2019 + 1) ⇒ (??, 2020)

snd (??, 2019 + 1) ⇒ 2020

(snd ??, 2019 + 1) ⇒ (snd ??, 2020)

Live Eval

$$e \Rightarrow r$$

$$r ::= \ldots \mid \lambda x.e \mid [E]\ ??_h \mid \text{snd } r \mid r_1 + r_2 \mid \ldots$$

Values

Paused Expressions

Live Eval

$$e \Rightarrow r$$

Live Uneval

$$r \Leftarrow ex \dashv X$$

$$r ::= \ldots \mid \lambda x.e \mid [E]\ ??_h \mid \mathbf{snd}\ r \mid r_1 + r_2 \mid \ldots$$

$$ex ::= \ldots \mid \{\, v_{in} \rightarrow ex_{out} \,\}$$

$$X ::= \ldots \mid X, (E \vdash ??_h \vDash ex)$$

Example constraints required by Myth

Live Uneval

$$r \Leftarrow ex \dashv X$$

$$
\begin{array}{rcl}
r & ::= & \ldots \mid \lambda x.e \mid [E]\ ??_h \mid \mathbf{snd}\ r \mid r_1 + r_2 \mid \ldots \\
ex & ::= & \ldots \mid \{\, v_{in} \rightarrow ex_{out} \,\} \\
X & ::= & \ldots \mid X, (E \vdash ??_h \vDash ex)
\end{array}
$$

$$
\begin{array}{rcl}
2020 & \Leftarrow & 2020 \dashv \emptyset \\
[E]\ ??_1 & \Leftarrow & 2020 \dashv (E \vdash ??_1 \vDash 2020) \\
4 + [E]\ ??_2 & \Leftarrow & 2020 \dashv (E \vdash ??_2 \vDash 2016) \\
\mathbf{ite}\ ([E]\ ??_3)\ 2016\ 2020 & \Leftarrow & 2020 \dashv (E \vdash ??_3 \vDash \mathbf{False})
\end{array}
$$

Live Eval

$$e \Rightarrow r$$

Live Uneval

$$r \Leftarrow ex \dashv X$$

$$
\begin{array}{rcl}
r & ::= & \ldots \mid \lambda x.e \mid [E]\ ??_h \mid \mathbf{snd}\ r \mid r_1 + r_2 \mid \ldots \\
ex & ::= & \ldots \mid \{\, v_{in} \rightarrow ex_{out} \,\} \\
X & ::= & \ldots \mid X, (E \vdash ??_h \vDash ex)
\end{array}
$$

$$E \vdash e \Rightarrow r \Leftarrow ex \dashv X$$

Live Bidirectional Evaluation

Live Eval

$$E \vdash e \Rightarrow r$$

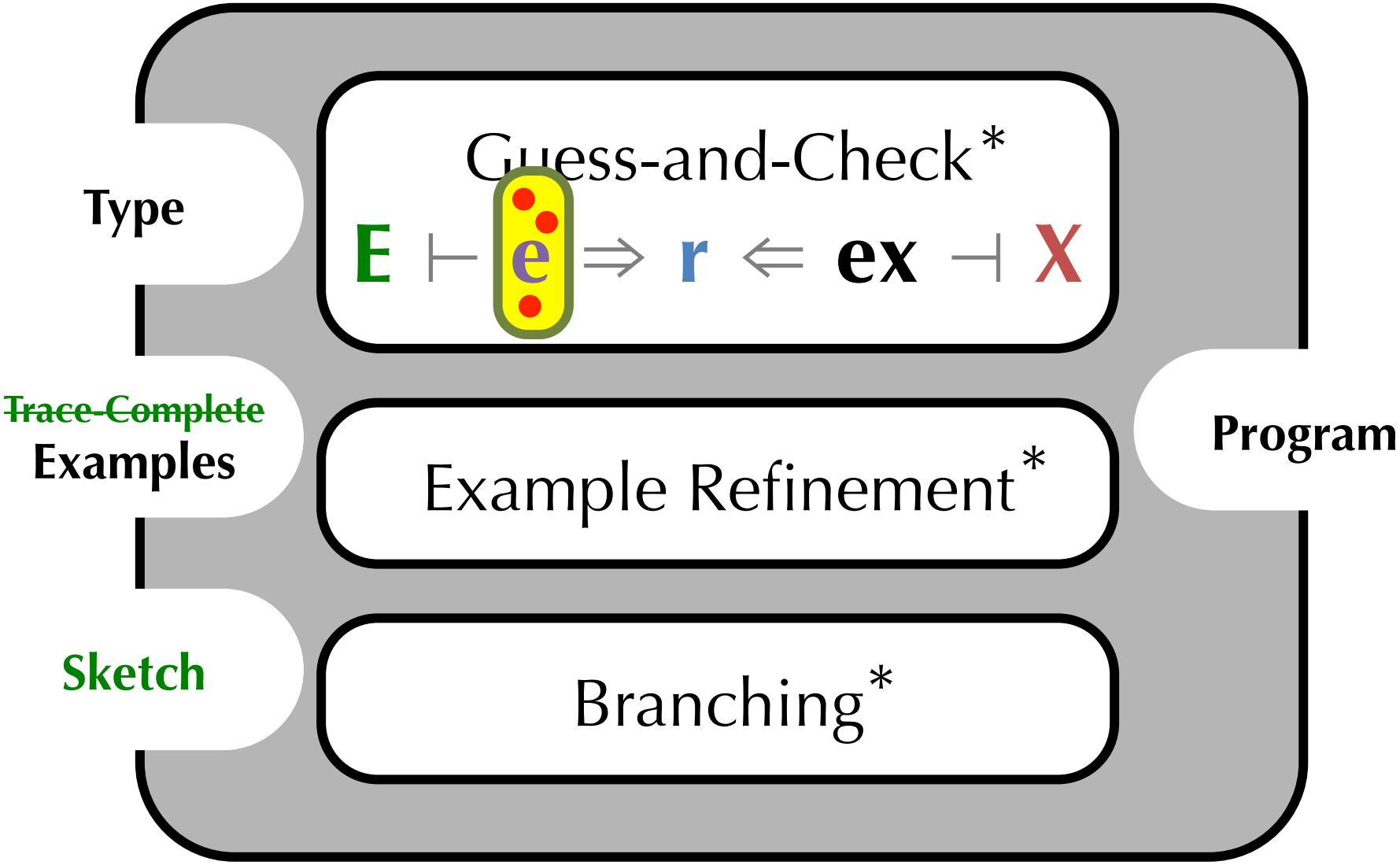
+

Live Uneval

$$r \Leftarrow ex \dashv X$$

Partially evaluate sketch to result r (value or "paused")

Constraints X s.t. r will resume and satisfy ex


Type
Guess-and-Check*
E ⊢ e ⇒ r ⇐ ex ⊣ X
~~Trace-Complete~~
Examples
Example Refinement*
Program
Sketch
Branching*

	Smyth					
Experiment	**1**		**2a**	**2b**	**3a**	**3b**
Sketch / Objective	*None / Top-1*		*None / Top-1*		*Base Case / Top-1-R*	
Name	**Expert**	**Time**	**Expert**	**Random** (50%, 90%)	**Expert**	**Random** (50%, 90%)
bool_band	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_bor	4	0.003	3 (75%)	(4,4)	•[3]	•[3]
bool_impl	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_neg	2	0.001	2 (100%)	(2,2)	•[3]	•[3]
bool_xor	4	0.009	4 (100%)	(4,4)	•[3]	•[3]
list_append	6	0.008	4 (67%)	(3,4)	1+1 (33%)	(1+3,1+4)
list_compress	13	timeout	•[1]	•[1]	•[1]	•[1]
list_concat	6	0.010	3 (50%)	(2,4)	incorrect	(1+3,1+5)
list_drop	11	0.092	5 (45%)	(6,9)	1+2 (27%)	(1+7,↓)
list_even_parity	7	overspec	•[1]	(–,–)	•[1]	(–,–)
list_filter	9*	0.144	5 (56%)	•[2]	1+4 (56%)	•[2]
list_fold	9	0.838	3 (33%)	•[2]	1+3 (44%)	•[2]
list_hd	3	0.003	2 (67%)	(2,3)	•[3]	•[3]
list_inc	4	0.018	2 (50%)	(2,2)	•[3]	•[3]
list_last	6	0.007	4 (67%)	(5,9)	1+2 (50%)	(1+5,1+10)
list_length	3	0.002	3 (100%)	(3,4)	1+1 (67%)	(1+2,1+2)
list_map	8	0.049	4 (50%)	•[2]	1+2 (38%)	•[2]
list_nth	13	0.124	5 (38%)	(7,14)	1+2 (23%)	(1+7,1+15)
list_pairwise_swap	7	0.634	5 (71%)	timeout	overspec	timeout
list_rev_append	5	0.107	3 (60%)	(5,8)	1+2 (60%)	(1+3,1+4)
list_rev_fold	5	0.035	2 (40%)	(2,4)	•[3]	•[3]
list_rev_snoc	5	0.010	3 (60%)	(3,6)	1+1 (40%)	(1+2,1+4)
list_rev_tailcall	8	0.008	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+5)
list_snoc	8	0.012	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+4)
list_sort_sorted_insert	7	0.015	3 (43%)	(3,6)	1+1 (29%)	(1+2,1+4)
list_sorted_insert	12	2.902	7 (58%)	timeout	1+7 (67%)	timeout
list_stutter	3	0.003	2 (67%)	(3,3)	1+1 (67%)	(1+2,1+3)
list_sum	3	0.029	2 (67%)	(2,2)	•[3]	•[3]
list_take	12	0.065	5 (42%)	(6,9)	1+3 (33%)	(1+7,1+16)
list_tl	3	0.002	2 (67%)	(2,3)	•[3]	•[3]
nat_add	9	0.006	4 (44%)	(5,6)	1+1 (22%)	(1+3,1+4)
nat_iseven	4	0.003	3 (75%)	(4,4)	1+2 (75%)	(1+3,1+4)
nat_max	9	0.041	9 (100%)	(8,12)	1+4 (56%)	(1+8,1+12)
nat_pred	3	0.001	2 (67%)	(2,3)	•[3]	•[3]
tree_binsert	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_collect_leaves	6	0.074	3 (50%)	$(3,4)^{t=3}$	1+2 (50%)	(1+3,1+3)
tree_count_leaves	7	2.660	3 (43%)	timeout	1+1 (29%)	timeout
tree_count_nodes	6	0.351	3 (50%)	$(4,\downarrow)^{t=10}$	1+2 (50%)	$(1+3,1+5)^{t=3}$
tree_inorder	5	0.123	4 (80%)	(3,4)	1+2 (60%)	(1+3,1+4)
tree_map	7	0.061	4 (57%)	•[2]	1+3 (57%)	•[2]
tree_nodes_at_level	11	timeout	•[1]	•[1]	•[1]	•[1]
tree_postorder	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_preorder	5	0.153	3 (60%)	$(3,4)^{t=3}$	1+2 (60%)	(1+3,1+3)
Averages			61%*		46%	

	Smyth					
Experiment	**1**		**2a**	**2b**	**3a**	**3b**
Sketch / Objective	*None / Top-1*		*None / Top-1*		*Base Case / Top-1-R*	
Name	**Expert**	**Time**	**Expert**	**Random** (50%, 90%)	**Expert**	**Random** (50%, 90%)
bool_band	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_bor	4	0.003	3 (75%)	(4,4)	•[3]	•[3]
bool_impl	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_neg	2	0.001	2 (100%)	(2,2)	•[3]	•[3]
bool_xor	4	0.009	4 (100%)	(4,4)	•[3]	•[3]
list_append	6	0.008	4 (67%)	(3,4)	1+1 (33%)	(1+3,1+4)
list_compress	13	timeout	•[1]	•[1]	•[1]	•[1]
list_concat	6	0.010	3 (50%)	(2,4)	incorrect	(1+3,1+5)
list_drop	11	0.092	5 (45%)	(6,9)	1+2 (27%)	(1+7,↓)
list_even_parity	7	overspec	•[1]	(–,–)	•[1]	(–,–)
list_filter	9*	0.144	5 (56%)	•[2]	1+4 (56%)	•[2]
list_fold	9	0.838	3 (33%)	•[2]	1+3 (44%)	•[2]
list_hd	3	0.003	2 (67%)	(2,3)	•[3]	•[3]
list_inc	4	0.018	2 (50%)	(2,2)	•[3]	•[3]
list_last	6	0.007	4 (67%)	(5,9)	1+2 (50%)	(1+5,1+10)
list_length	3	0.002	3 (100%)	(3,4)	1+1 (67%)	(1+2,1+2)
list_map	8	0.049	4 (50%)	•[2]	1+2 (38%)	•[2]
list_nth	13	0.12	5 (38%)	timeout	1+3 (23%)	timeout
list_pairwise_swap	7	0.834	5 (71%)	timeout	overspec	timeout
list_rev_append	5	0.107	3 (60%)	(5,8)	1+2 (60%)	(1+3,1+4)
list_rev_fold	5	0.035	2 (40%)	(2,4)	•[3]	•[3]
list_rev_snoc	5	0.010	3 (60%)	(3,6)	1+1 (40%)	(1+2,1+4)
list_rev_tailcall	8	0.008	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+5)
list_snoc	8	0.012	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+4)
list_sort_sorted_insert	7	0.015	3 (43%)	(3,6)	1+1 (29%)	(1+2,1+4)
list_sorted_insert	12	2.902	7 (58%)	timeout	1+7 (67%)	timeout
list_stutter	3	0.003	2 (67%)	(3,3)	1+1 (67%)	(1+2,1+3)
list_sum	3	0.029	2 (67%)	(2,2)	•[3]	•[3]
list_take	12	0.065	5 (42%)	(6,9)	1+3 (33%)	(1+7,1+16)
list_tl	3	0.002	2 (67%)	(2,3)	•[3]	•[3]
nat_add	9	0.006	4 (44%)	(5,6)	1+1 (22%)	(1+3,1+4)
nat_iseven	4	0.003	3 (75%)	(4,4)	1+2 (75%)	(1+3,1+4)
nat_max	9	0.041	9 (100%)	(8,12)	1+4 (56%)	(1+8,1+12)
nat_pred	3	0.001	2 (67%)	(2,3)	•[3]	•[3]
tree_binsert	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_collect_leaves	6	0.074	3 (50%)	$(3,4)^{t=3}$	1+2 (50%)	(1+3,1+3)
tree_count_leaves	7	2.660	3 (43%)	timeout	1+1 (29%)	timeout
tree_count_nodes	6	0.351	3 (50%)	$(4,\downarrow)^{t=10}$	1+2 (50%)	$(1+3,1+5)^{t=3}$
tree_inorder	5	0.123	4 (80%)	(3,4)	1+2 (60%)	(1+3,1+4)
tree_map	7	0.061	4 (57%)	•[2]	1+3 (57%)	•[2]
tree_nodes_at_level	11	timeout	•[1]	•[1]	•[1]	•[1]
tree_postorder	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_preorder	5	0.153	3 (60%)	$(3,4)^{t=3}$	1+2 (60%)	(1+3,1+3)
Averages			61%*		46%	

Compared to prior state-of-the-art:

Trace-complete examples (without a sketch)

	Smyth					
Experiment	**1**		**2a**	**2b**	**3a**	**3b**
Sketch / Objective	*None / Top-1*		*None / Top-1*		*Base Case / Top-1-R*	
Name	**Expert**	**Time**	**Expert**	**Random** (50%, 90%)	**Expert**	**Random** (50%, 90%)
bool_band	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_bor	4	0.003	3 (75%)	(4,4)	•[3]	•[3]
bool_impl	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_neg	2	0.001	2 (100%)	(2,2)	•[3]	•[3]
bool_xor	4	0.009	4 (100%)	(4,4)	•[3]	•[3]
list_append	6	0.008	4 (67%)	(3,4)	1+1 (33%)	(1+3,1+4)
list_compress	13	timeout	•[1]	•[1]	•[1]	•[1]
list_concat	6	0.010	3 (50%)	(2,4)	incorrect	(1+3,1+5)
list_drop	11	0.092	5 (45%)	(6,9)	1+2 (27%)	(1+7,↓)
list_even_parity	7	overspec	•[1]	(–,–)	•[1]	(–,–)
list_filter	9*	0.144	5 (56%)	•[2]	1+4 (56%)	•[2]
list_fold	9	0.838	3 (33%)	•[2]	1+3 (44%)	•[2]
list_hd	3	0.003	2 (67%)	(2,3)	•[3]	•[3]
list_inc	4	0.018	2 (50%)	(2,2)	•[3]	•[3]
list_last	6	0.007	4 (67%)	(5,9)	1+2 (50%)	(1+5,1+10)
list_length	3	0.002	3 (100%)	(3,4)	1+1 (67%)	(1+2,1+2)
list_map	8	0.049	4 (50%)	•[2]	1+2 (38%)	•[2]
list_nth	13	0.124	5 (38%)	(7,14)	1+2 (23%)	(1+7,1+15)
list_pairwise_swap	7	0.634	5 (71%)	timeout	overspec	timeout
list_rev_append	5	0.107	3 (60%)	(5,8)	1+2 (60%)	(1+3,1+4)
list_rev_fold	5	0.035	2 (40%)	(2,4)	•[3]	•[3]
list_rev_snoc	5	0.010	3 (60%)	(3,6)	1+1 (40%)	(1+2,1+4)
list_rev_tailcall	8	0.008	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+5)
list_snoc	8	0.012	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+4)
list_sort_sorted_insert	7	0.015	3 (43%)	(3,6)	1+1 (29%)	(1+2,1+4)
list_sorted_insert	12	2.902	7 (58%)	timeout	1+7 (67%)	timeout
list_stutter	3	0.003	2 (67%)	(3,3)	1+1 (67%)	(1+2,1+3)
list_sum	3	0.029	2 (67%)	(2,2)	•[3]	•[3]
list_take	12	0.065	5 (42%)	(6,9)	1+3 (33%)	(1+7,1+16)
list_tl	3	0.002	2 (67%)	(2,3)	•[3]	•[3]
nat_add	9	0.006	4 (44%)	(5,6)	1+1 (22%)	(1+3,1+4)
nat_iseven	4	0.003	3 (75%)	(4,4)	1+2 (75%)	(1+3,1+4)
nat_max	9	0.041	9 (100%)	(8,12)	1+4 (56%)	(1+8,1+12)
nat_pred	3	0.001	2 (67%)	(2,3)	•[3]	•[3]
tree_binsert	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_collect_leaves	6	0.074	3 (50%)	$(3,4)^{t=3}$	1+2 (50%)	(1+3,1+3)
tree_count_leaves	7	2.660	3 (43%)	timeout	1+1 (29%)	timeout
tree_count_nodes	6	0.351	3 (50%)	$(4,\downarrow)^{t=10}$	1+2 (50%)	$(1+3,1+5)^{t=3}$
tree_inorder	5	0.123	4 (80%)	(3,4)	1+2 (60%)	(1+3,1+4)
tree_map	7	0.061	4 (57%)	•[2]	1+3 (57%)	•[2]
tree_nodes_at_level	11	timeout	•[1]	•[1]	•[1]	•[1]
tree_postorder	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_preorder	5	0.153	3 (60%)	$(3,4)^{t=3}$	1+2 (60%)	(1+3,1+3)
Averages			61%*		46%	

Compared to prior state-of-the-art:

Non-trace-complete examples reduce specification size

	Smyth					
Experiment	**1**		**2a**	**2b**	**3a**	**3b**
Sketch / Objective	*None / Top-1*		*None / Top-1*		[illegible] / *Top-1-R*	
Name	**Expert**	**Time**	**Expert**	**Random** (50%, 90%)	**Expert**	**Random** (50%, 90%)
bool_band	4	0.004	3 (75%)	(4,4)	$\bullet^{3}$	$\bullet^{3}$
bool_bor	4	0.003	3 (75%)	(4,4)	$\bullet^{3}$	$\bullet^{3}$
bool_impl	4	0.004	3 (75%)	(4,4)	$\bullet^{3}$	$\bullet^{3}$
bool_neg	2	0.001	2 (100%)	(2,2)	$\bullet^{3}$	$\bullet^{3}$
bool_xor	4	0.009	4 (100%)	(4,4)	$\bullet^{3}$	$\bullet^{3}$
list_append	6	0.008	4 (67%)	(3,4)	1+1 (33%)	(1+3,1+4)
list_compress	13	timeout	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$
list_concat	6	0.010	3 (50%)	(2,4)	incorrect	(1+3,1+5)
list_drop	11	0.092	5 (45%)	(6,9)	1+2 (27%)	(1+7,↓)
list_even_parity	7	overspec	$\bullet^{1}$	(–,–)	$\bullet^{1}$	(–,–)
list_filter	9*	0.144	5 (56%)	$\bullet^{2}$	1+4 (56%)	$\bullet^{2}$
list_fold	9	0.838	3 (33%)	$\bullet^{2}$	1+3 (44%)	$\bullet^{2}$
list_hd	3	0.003	2 (67%)	(2,3)	$\bullet^{3}$	$\bullet^{3}$
list_inc	4	0.018	2 (50%)	(2,2)	$\bullet^{3}$	$\bullet^{3}$
list_last	6	0.007	4 (67%)	(5,9)	1+2 (50%)	(1+5,1+10)
list_length	3	0.002	3 (100%)	(3,4)	1+1 (67%)	(1+2,1+2)
list_map	8	0.049	4 (50%)	$\bullet^{2}$	1+2 (38%)	$\bullet^{2}$
list_nth	13	0.124	5 (38%)	(7,14)	1+2 (23%)	(1+7,1+15)
list_pairwise_swap	7	0.634	5 (71%)	timeout	overspec	timeout
list_rev_append	5	0.107	3 (60%)	(5,8)	1+2 (60%)	(1+3,1+4)
list_rev_fold	5	0.035	2 (40%)	(2,4)	$\bullet^{3}$	$\bullet^{3}$
list_rev_snoc	5	0.010	3 (60%)	(3,6)	1+1 (40%)	(1+2,1+4)
list_rev_tailcall	8	0.008	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+5)
list_snoc	8	0.012	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+4)
list_sort_sorted_insert	7	0.015	3 (43%)	(3,6)	1+1 (29%)	(1+2,1+4)
list_sorted_insert	12	2.902	7 (58%)	timeout	1+7 (67%)	timeout
list_stutter	3	0.003	2 (67%)	(3,3)	1+1 (67%)	(1+2,1+3)
list_sum	3	0.029	2 (67%)	(2,2)	$\bullet^{3}$	$\bullet^{3}$
list_take	12	0.065	5 (42%)	(6,9)	1+3 (33%)	(1+7,1+16)
list_tl	3	0.002	2 (67%)	(2,3)	$\bullet^{3}$	$\bullet^{3}$
nat_add	9	0.006	4 (44%)	(5,6)	1+1 (22%)	(1+3,1+4)
nat_iseven	4	0.003	3 (75%)	(4,4)	1+2 (75%)	(1+3,1+4)
nat_max	9	0.041	9 (100%)	(8,12)	1+4 (56%)	(1+8,1+12)
nat_pred	3	0.001	2 (67%)	(2,3)	$\bullet^{3}$	$\bullet^{3}$
tree_binsert	20	timeout	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$
tree_collect_leaves	6	0.074	3 (50%)	$(3,4)^{t=3}$	1+2 (50%)	(1+3,1+3)
tree_count_leaves	7	2.660	3 (43%)	timeout	1+1 (29%)	timeout
tree_count_nodes	6	0.351	3 (50%)	$(4,\downarrow)^{t=10}$	1+2 (50%)	$(1+3,1+5)^{t=3}$
tree_inorder	5	0.123	4 (80%)	(3,4)	1+2 (60%)	(1+3,1+4)
tree_map	7	0.061	4 (57%)	$\bullet^{2}$	1+3 (57%)	$\bullet^{2}$
tree_nodes_at_level	11	timeout	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$
tree_postorder	20	timeout	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$	$\bullet^{1}$
tree_preorder	5	0.153	3 (60%)	$(3,4)^{t=3}$	1+2 (60%)	(1+3,1+3)
Averages			61%*		46%	

Compared to prior state-of-the-art:

Non-trace-complete examples reduce specification size

Sketches reduce specification size

	Smyth					
Experiment	**1**		**2a**	**2b**	**3a**	**3b**
Sketch / Objective	*None / Top-1*		*None*		*Base Co*	
Name	**Expert**	**Time**	**Expert**	**Random** (50%, 90%)	**Expert**	**Random** (50%, 90%)
bool_band	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_bor	4	0.003	3 (75%)	(4,4)	•[3]	•[3]
bool_impl	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_neg	2	0.001	2 (100%)	(2,2)	•[3]	•[3]
bool_xor	4	0.009	4 (100%)	(4,4)	•[3]	•[3]
list_append	6	0.008	4 (67%)	(3,4)	1+1 (33%)	(1+3,1+4)
list_compress	13	timeout	•[1]	•[1]	•[1]	•[1]
list_concat	6	0.010	3 (50%)	(2,4)	incorrect	(1+3,1+5)
list_drop	11	0.092	5 (45%)	(6,9)	1+2 (27%)	(1+7,↓)
list_even_parity	7	overspec	•[1]	(–,–)	•[1]	(–,–)
list_filter	9*	0.144	5 (56%)	•[2]	1+4 (56%)	•[2]
list_fold	9	0.838	3 (33%)	•[2]	1+3 (44%)	•[2]
list_hd	3	0.003	2 (67%)	(2,3)	•[3]	•[3]
list_inc	4	0.018	2 (50%)	(2,2)	•[3]	•[3]
list_last	6	0.007	4 (67%)	(5,9)	1+2 (50%)	(1+5,1+10)
list_length	3	0.002	3 (100%)	(3,4)	1+1 (67%)	(1+2,1+2)
list_map	8	0.049	4 (50%)	•[2]	1+2 (38%)	•[2]
list_nth	13	0.124	5 (38%)	(7,14)	1+2 (23%)	(1+7,1+15)
list_pairwise_swap	7	0.634	5 (71%)	timeout	overspec	timeout
list_rev_append	5	0.107	3 (60%)	(5,8)	1+2 (60%)	(1+3,1+4)
list_rev_fold	5	0.035	2 (40%)	(2,4)	•[3]	•[3]
list_rev_snoc	5	0.010	3 (60%)	(3,6)	1+1 (40%)	(1+2,1+4)
list_rev_tailcall	8	0.008	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+5)
list_snoc	8	0.012	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+4)
list_sort_sorted_insert	7	0.015	3 (43%)	(3,6)	1+1 (29%)	(1+2,1+4)
list_sorted_insert	12	2.902	7 (58%)	timeout	1+7 (67%)	timeout
list_stutter	3	0.003	2 (67%)	(3,3)	1+1 (67%)	(1+2,1+3)
list_sum	3	0.029	2 (67%)	(2,2)	•[3]	•[3]
list_take	12	0.065	5 (42%)	(6,9)	1+3 (33%)	(1+7,1+16)
list_tl	3	0.002	2 (67%)	(2,3)	•[3]	•[3]
nat_add	9	0.006	4 (44%)	(5,6)	1+1 (22%)	(1+3,1+4)
nat_iseven	4	0.003	3 (75%)	(4,4)	1+2 (75%)	(1+3,1+4)
nat_max	9	0.041	9 (100%)	(8,12)	1+4 (56%)	(1+8,1+12)
nat_pred	3	0.001	2 (67%)	(2,3)	•[3]	•[3]
tree_binsert	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_collect_leaves	6	0.074	3 (50%)	$(3,4)^{t=3}$	1+2 (50%)	(1+3,1+3)
tree_count_leaves	7	2.660	3 (43%)	timeout	1+1 (29%)	timeout
tree_count_nodes	6	0.351	3 (50%)	$(4,\downarrow)^{t=10}$	1+2 (50%)	$(1+3,1+5)^{t=3}$
tree_inorder	5	0.123	4 (80%)	(3,4)	1+2 (60%)	(1+3,1+4)
tree_map	7	0.061	4 (57%)	•[2]	1+3 (57%)	•[2]
tree_nodes_at_level	11	timeout	•[1]	•[1]	•[1]	•[1]
tree_postorder	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_preorder	5	0.153	3 (60%)	$(3,4)^{t=3}$	1+2 (60%)	(1+3,1+3)
Averages			61%*		46%	

Compared to prior state-of-the-art:

Non-trace-complete examples reduce specification size

Sketches reduce specification size

Synthesis (relatively) ***robust*** to choice of examples

	Smyth					
Experiment	**1**		**2a**	**2b**	**3a**	**3b**
Sketch / Objective	*None / Top-1*		*None / Top-1*		*Base Case / Top-1-R*	
Name	**Expert**	**Time**	**Expert**	**Random** (50%, 90%)	**Expert**	**Random** (50%, 90%)
bool_band	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_bor	4	0.003	3 (75%)	(4,4)	•[3]	•[3]
bool_impl	4	0.004	3 (75%)	(4,4)	•[3]	•[3]
bool_neg	2	0.001	2 (100%)	(2,2)	•[3]	•[3]
bool_xor	4	0.009	4 (100%)	(4,4)	•[3]	•[3]
list_append	6	0.008	4 (67%)	(3,4)	1+1 (33%)	(1+3,1+4)
list_compress	13	timeout	•[1]	•[1]	•[1]	•[1]
list_concat	6	0.010	3 (50%)	(2,4)	incorrect	(1+3,1+5)
list_drop	11	0.092	5 (45%)	(6,9)	1+2 (27%)	(1+7,↓)
list_even_parity	7	overspec	•[1]	(–,–)	•[1]	(–,–)
list_filter	9*	0.144	5 (56%)	•[2]	1+4 (56%)	•[2]
list_fold	9	0.838	3 (33%)	•[2]	1+3 (44%)	•[2]
list_hd	3	0.003	2 (67%)	(2,3)	•[3]	•[3]
list_inc	4	0.018	2 (50%)	(2,2)	•[3]	•[3]
list_last	6	0.007	4 (67%)	(5,9)	1+2 (50%)	(1+5,1+10)
list_length	3	0.002	3 (100%)	(3,4)	1+1 (67%)	(1+2,1+2)
list_map	8	0.049	4 (50%)	•[2]	1+2 (38%)	•[2]
list_nth	13	0.124	5 (38%)	(7,14)	1+2 (23%)	(1+7,1+15)
list_pairwise_swap	7	0.634	5 (71%)	timeout	overspec	timeout
list_rev_append	5	0.107	3 (60%)	(5,8)	1+2 (60%)	(1+3,1+4)
list_rev_fold	5	0.035	2 (40%)	(2,4)	•[3]	•[3]
list_rev_snoc	5	0.010	3 (60%)	(3,6)	1+1 (40%)	(1+2,1+4)
list_rev_tailcall	8	0.008	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+5)
list_snoc	8	0.012	3 (38%)	(3,4)	1+1 (25%)	(1+3,1+4)
list_sort_sorted_insert	7	0.015	3 (43%)	(3,6)	1+1 (29%)	(1+2,1+4)
list_sorted_insert	12	2.902	7 (58%)	timeout	1+7 (67%)	timeout
list_stutter	3	0.003	2 (67%)	(3,3)	1+1 (67%)	(1+2,1+3)
list_sum	3	0.029	2 (67%)	(2,2)	•[3]	•[3]
list_take	12	0.065	5 (42%)	(6,9)	1+3 (33%)	(1+7,1+16)
list_tl	3	0.002	2 (67%)	(2,3)	•[3]	•[3]
nat_add	9	0.006	4 (44%)	(5,6)	1+1 (22%)	(1+3,1+4)
nat_iseven	4	0.003	3 (75%)	(4,4)	1+2 (75%)	(1+3,1+4)
nat_max	9	0.041	9 (100%)	(8,12)	1+4 (56%)	(1+8,1+12)
nat_pred	3	0.001	2 (67%)	(2,3)	•[3]	•[3]
tree_binsert	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_collect_leaves	6	0.074	3 (50%)	$(3,4)^{t=3}$	1+2 (50%)	(1+3,1+3)
tree_count_leaves	7	2.660	3 (43%)	timeout	1+1 (29%)	timeout
tree_count_nodes	6	0.351	3 (50%)	$(4,\downarrow)^{t=10}$	1+2 (50%)	$(1+3,1+5)^{t=3}$
tree_inorder	5	0.123	4 (80%)	(3,4)	1+2 (60%)	(1+3,1+4)
tree_map	7	0.061	4 (57%)	•[2]	1+3 (57%)	•[2]
tree_nodes_at_level	11	timeout	•[1]	•[1]	•[1]	•[1]
tree_postorder	20	timeout	•[1]	•[1]	•[1]	•[1]
tree_preorder	5	0.153	3 (60%)	$(3,4)^{t=3}$	1+2 (60%)	(1+3,1+3)
Averages			61%*		46%	

Compared to prior state-of-the-art:

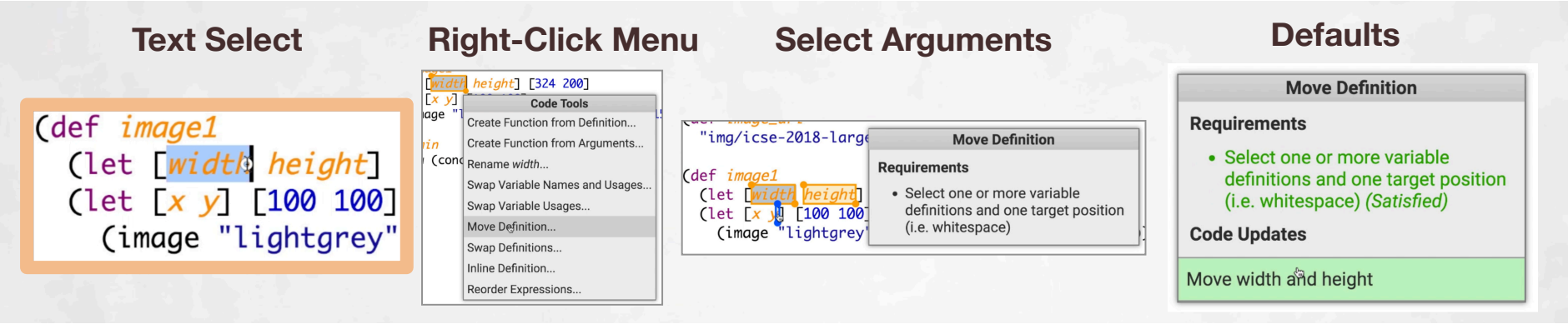
Non-trace-complete examples reduce specification size

Sketches reduce specification size

Synthesis (relatively) ***robust*** to choice of examples

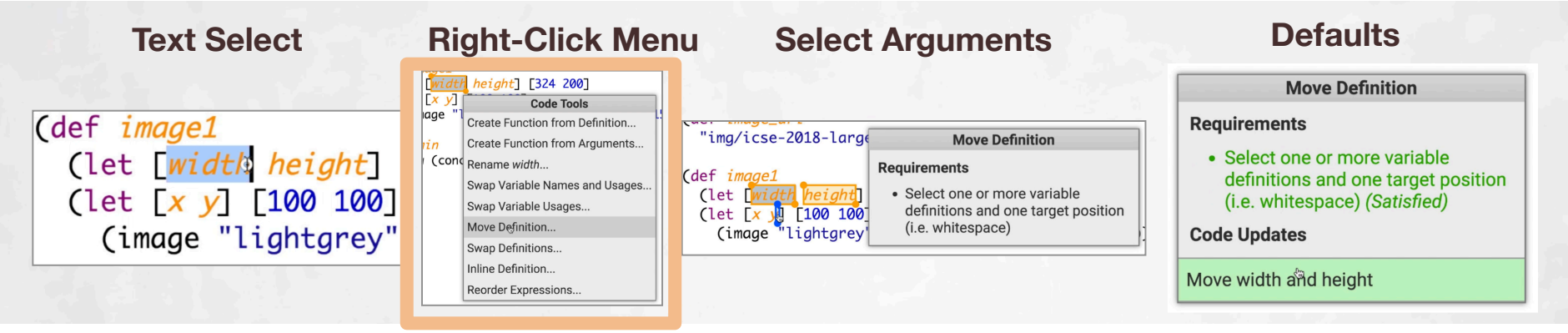
Hybrid Text-GUI Code Editors

Traditional Refactoring UI



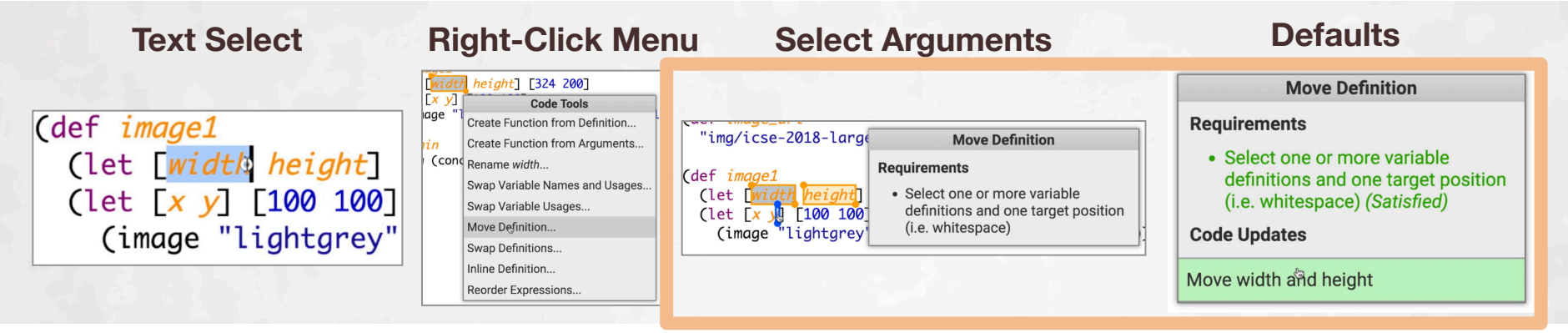
Text-Select

Traditional Refactoring UI



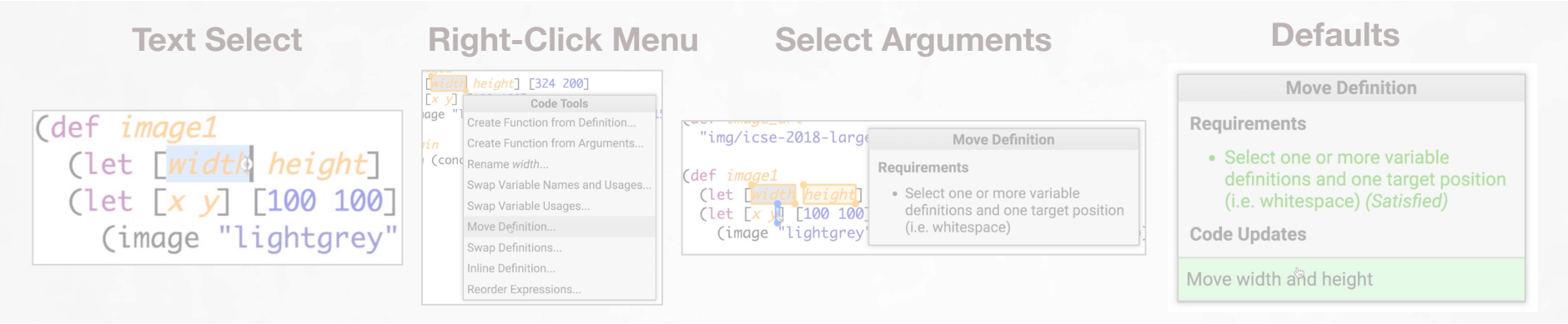
Text-Select + Long Menu

Traditional Refactoring UI



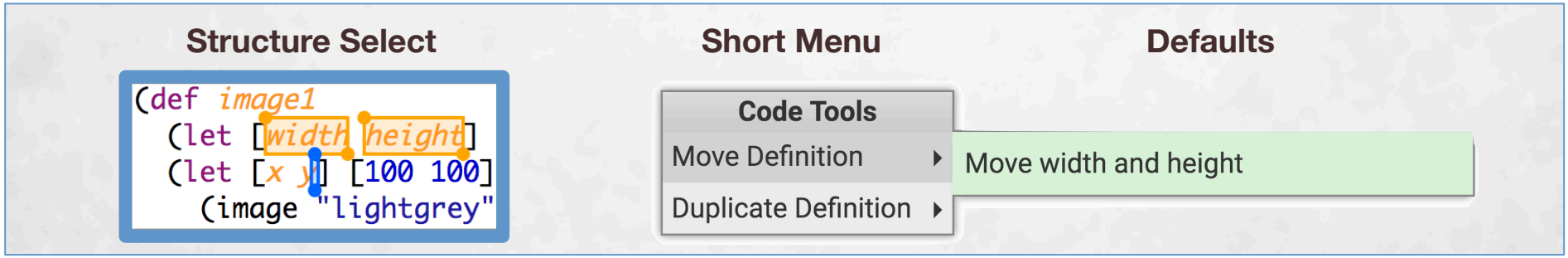
Text-Select + Long Menu + Configuration Choices

Traditional Refactoring UI



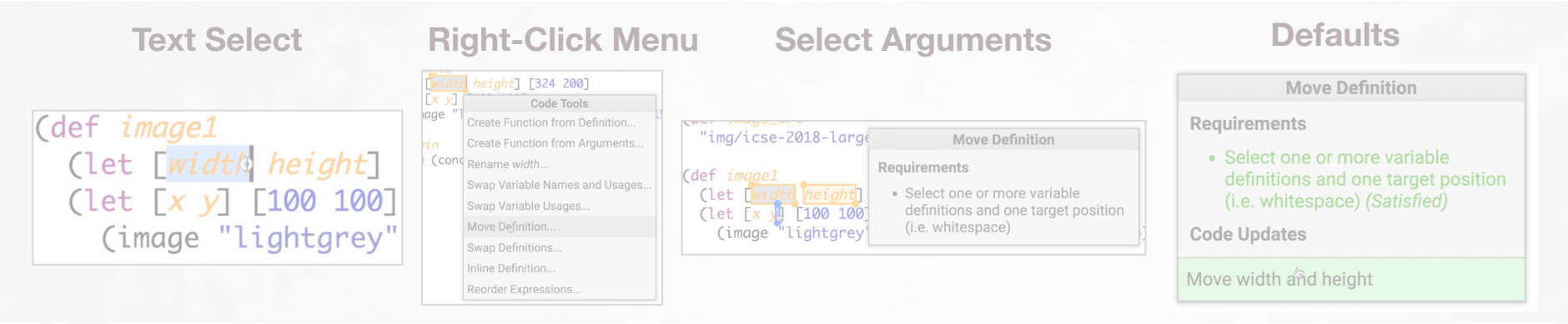
Text-Select + Long Menu + Configuration Choices

Deuce Refactoring UI



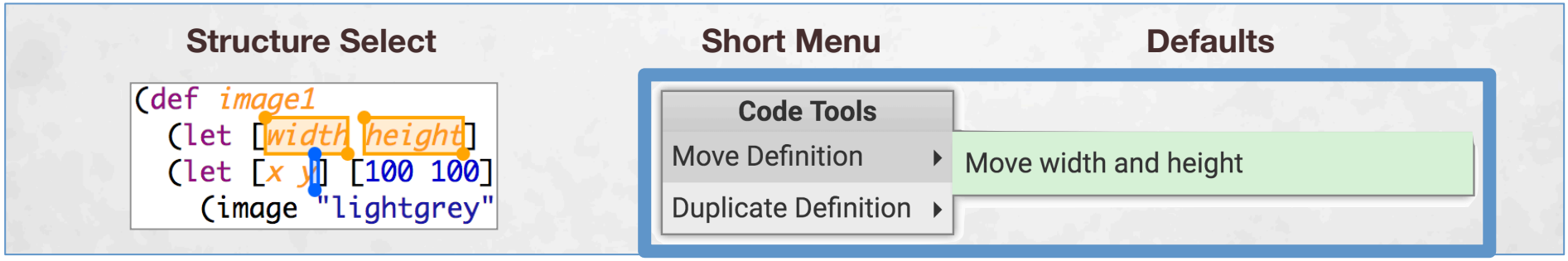
Structural Multi-Select

Traditional Refactoring UI

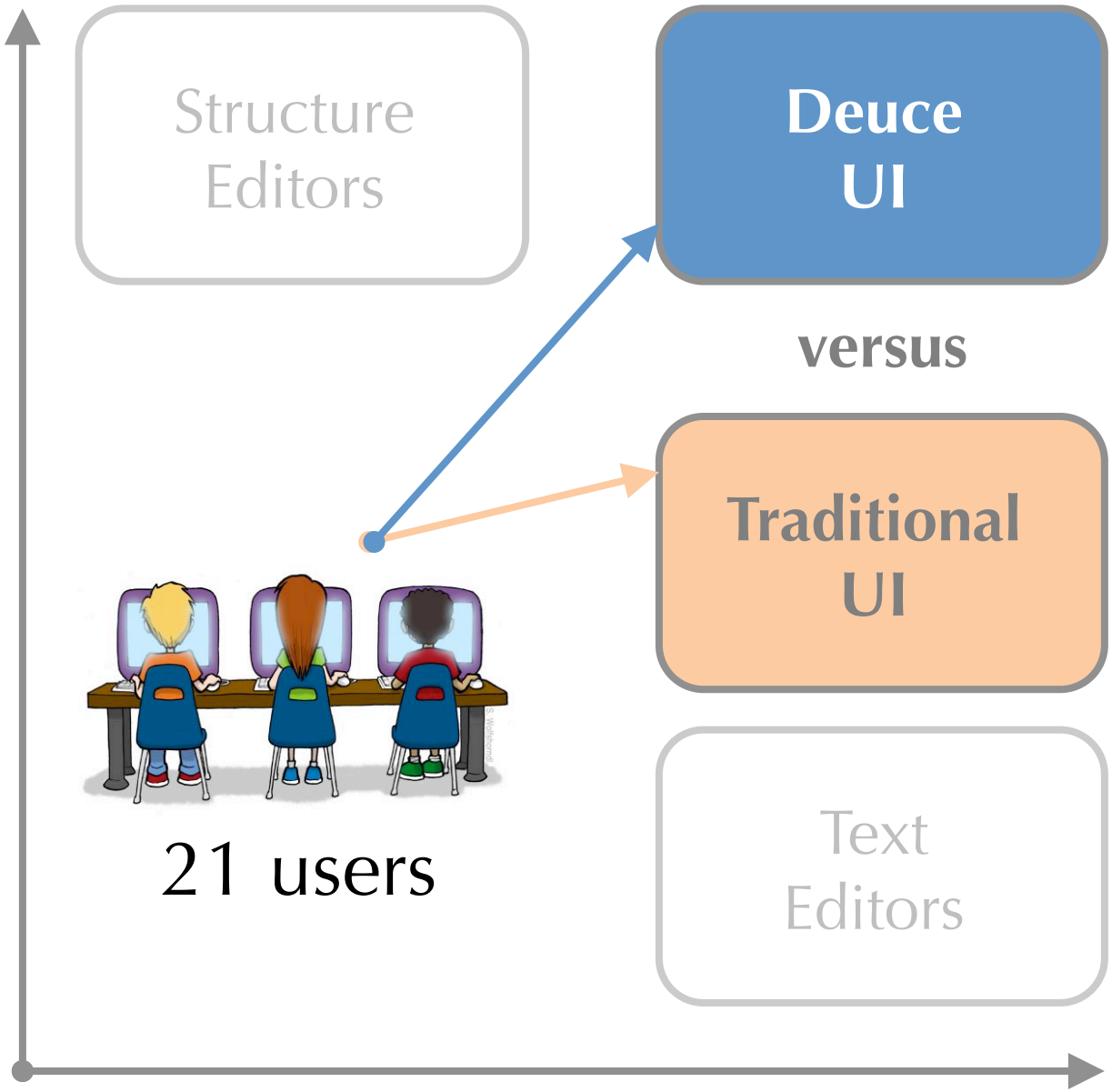


Text-Select + Long Menu + Configuration Choices

Deuce Refactoring UI



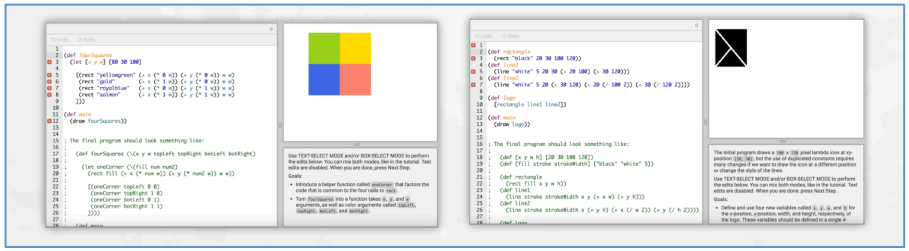
Structural Multi-Select + Context-Sensitive Transforms


Structure Editors
Deuce UI
versus
Traditional UI
21 users
Text Editors

1. Tutorial

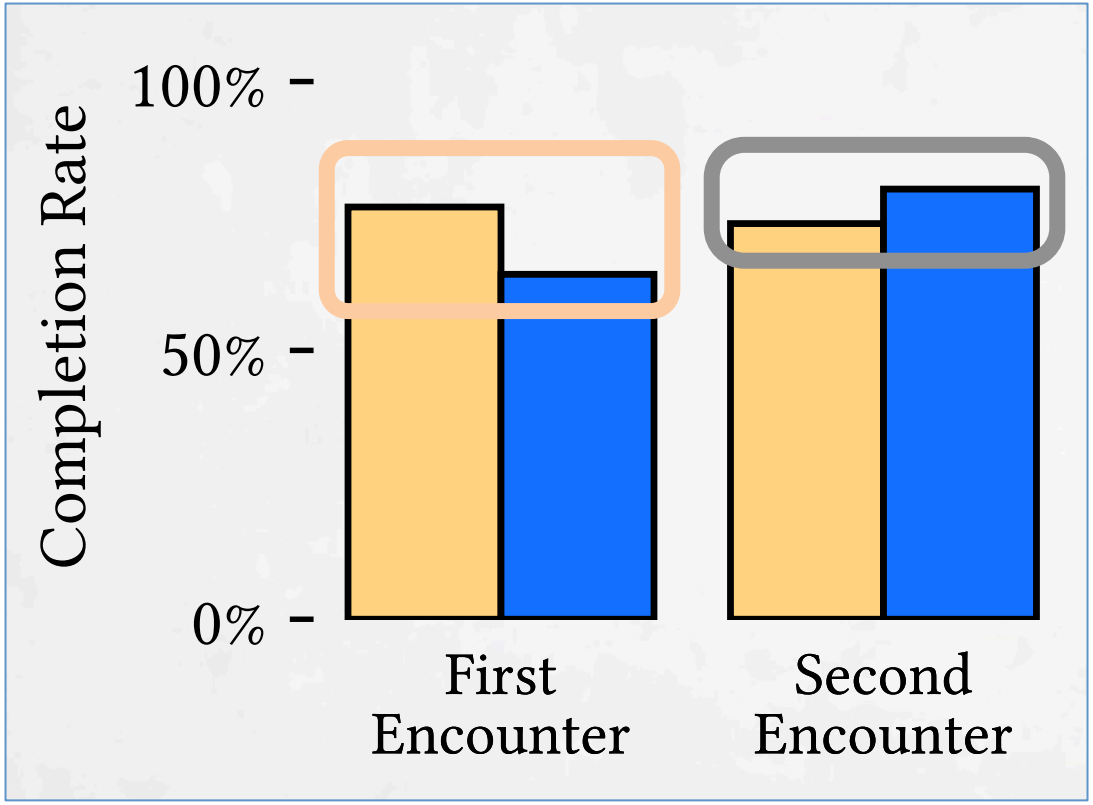
2. Head-to-Head Tasks (2x each; once per mode)

3. Mix-and-Match Tasks (free to use both modes)



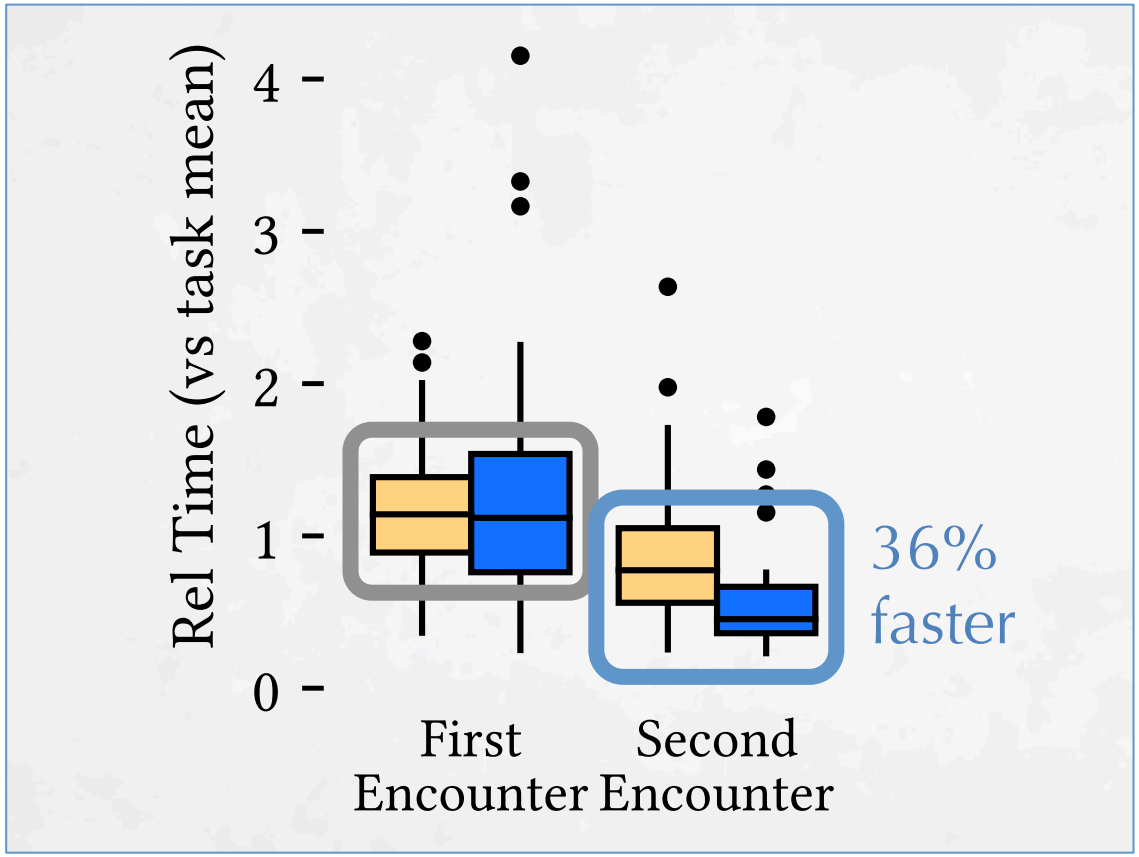
4. Exit Survey

Deuce more effective than Traditional?



Deuce doesn't help discoverability

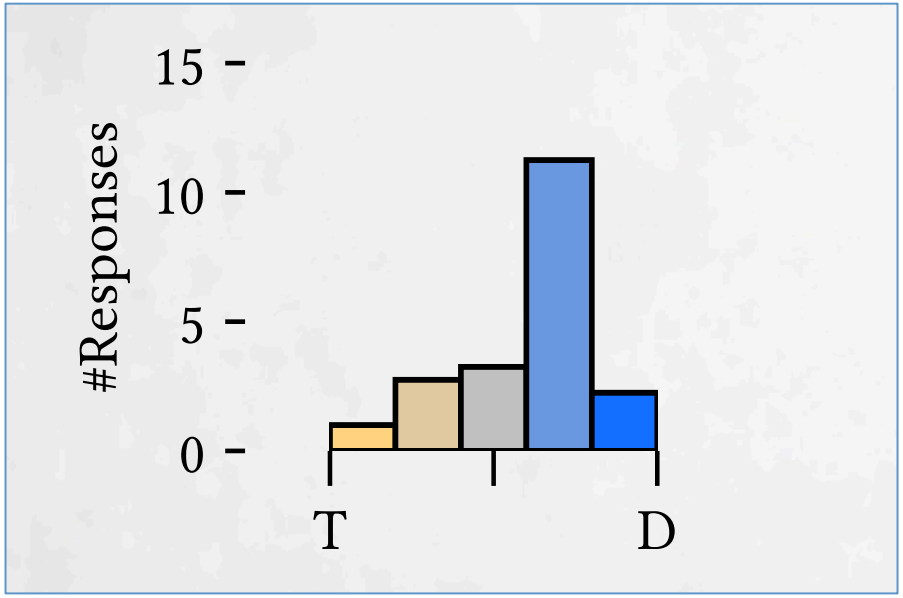
Deuce more effective than Traditional?



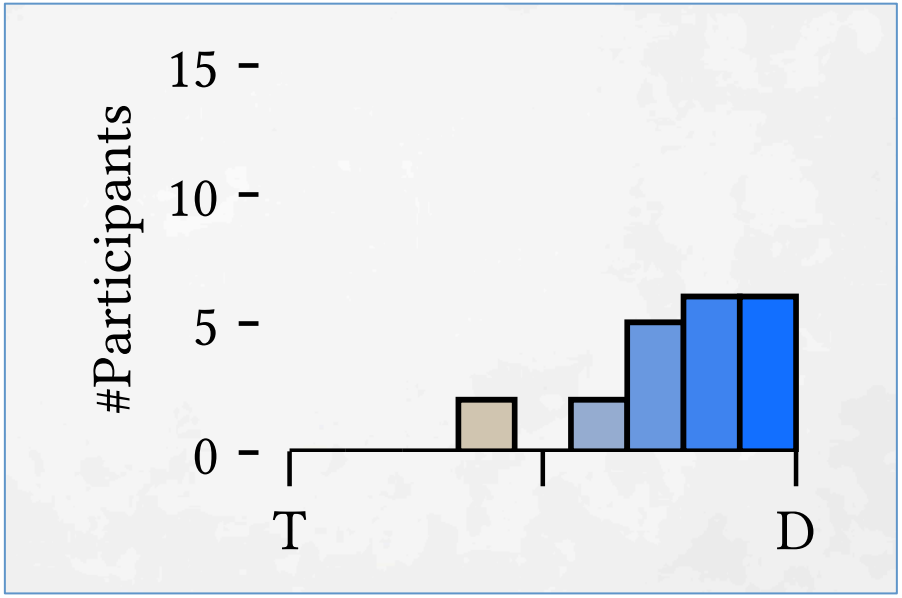
Deuce may be faster once learned

Deuce preferred to Traditional?

Survey



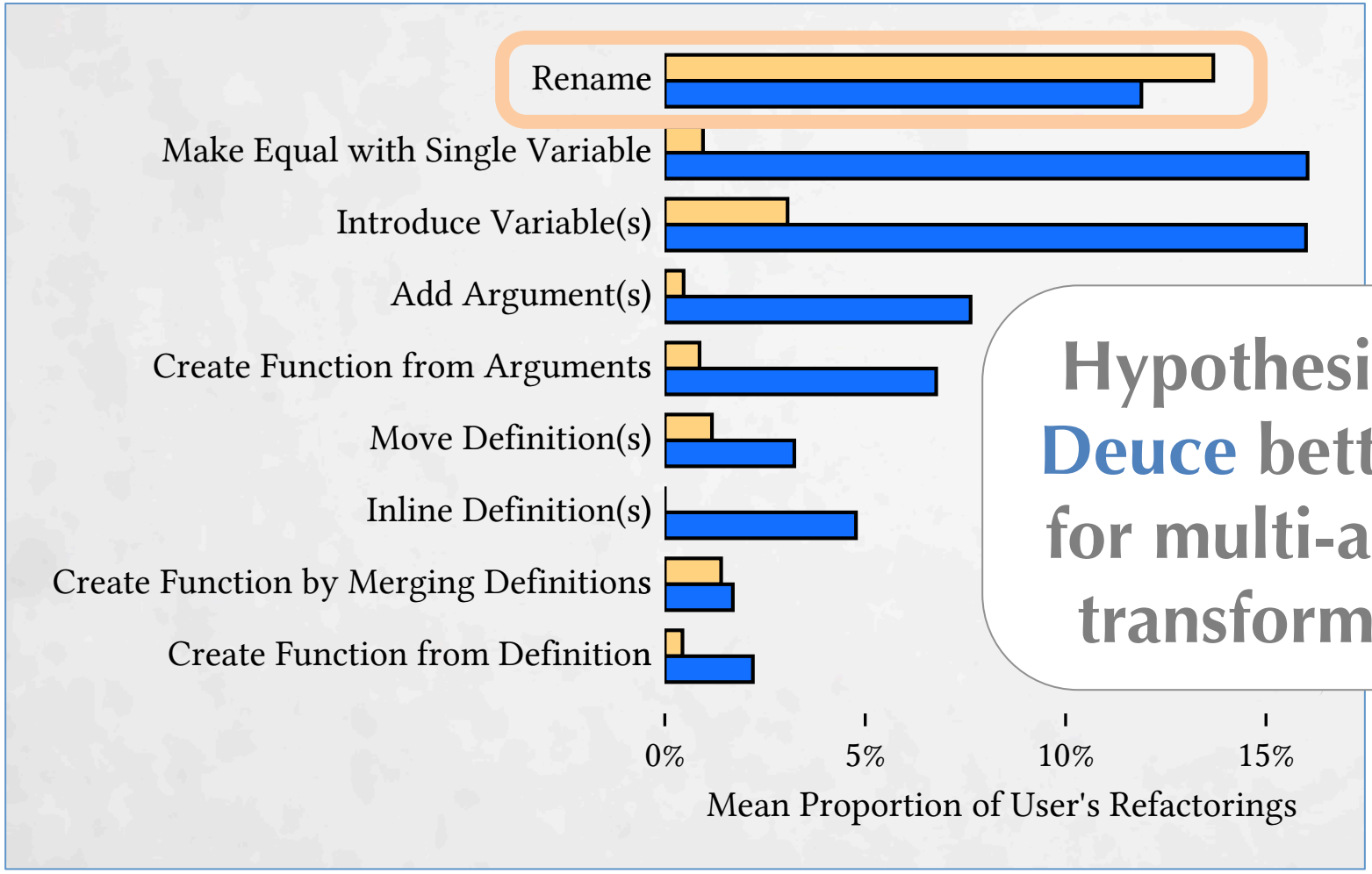
Observed



Modest subjective preference for Deuce

Almost everyone used Deuce more

Mix-and-Match Tool Usage



Hypothesis: Deuce better for multi-arg transforms

Deuce UI versus **Traditional UI**

Traditional may be better for discovery

Deuce may be faster once learned

Deuce strongly preferred

Future Work

Bidirectional Evaluation

Future Work

Synthesis of larger structural repairs

More expressive value diffs

Automatically deriving unevaluators

Bi-directional, scriptable
Docs, Sheets, Slides, p5.js, D3, Vega, etc.

Live Programming by Example

Future Work

Unpredictability of examples

Synthesizing pattern and type holes

Integrating logic-based specifications

Propagating multiple candidate programs ("multi-sketches")

Output-Directed Programming

Future Work

Propagating multiple candidate programs ("multi-sketches")

Visualization of intermediate program states

Deriving structure editors from `toString` and `toSvg` functions

DSLs for defining new transformations

Hybrid Text-GUI Code Editors

Future Work

DSLs for defining new transformations

DSLs for defining custom, type-specific GUIs

UI concerns for larger programs

How to encourage tree transforms?

Usable Synthesis

Future Work

Explaining program differences

Statistics to rank solutions according to code corpus and edit history

Representation of search space and intermediate synthesis results

Interaction to resolve ambiguity

END