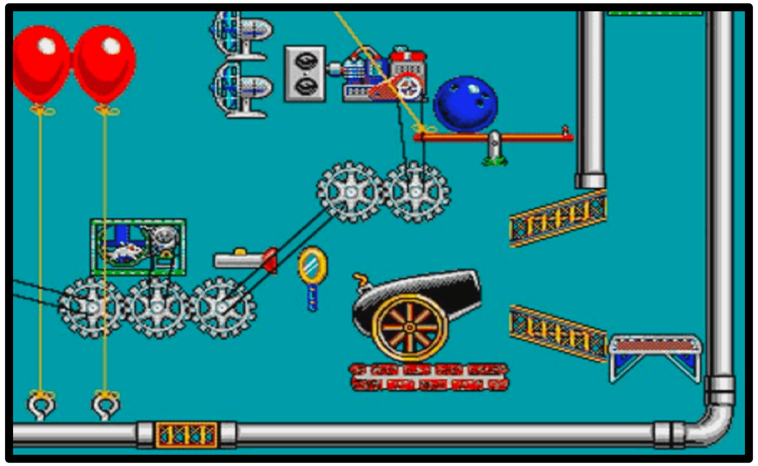
Bottom-up Synthesis of Memory Mutations with Separation Logic

Kasra Ferdowsi, Hila Peleg

Programming by Example

$\text{input}_1 \rightarrow \text{output}_1$
$\text{input}_2 \rightarrow \text{output}_2$
$\text{input}_3 \rightarrow \text{output}_3$

$\Downarrow$

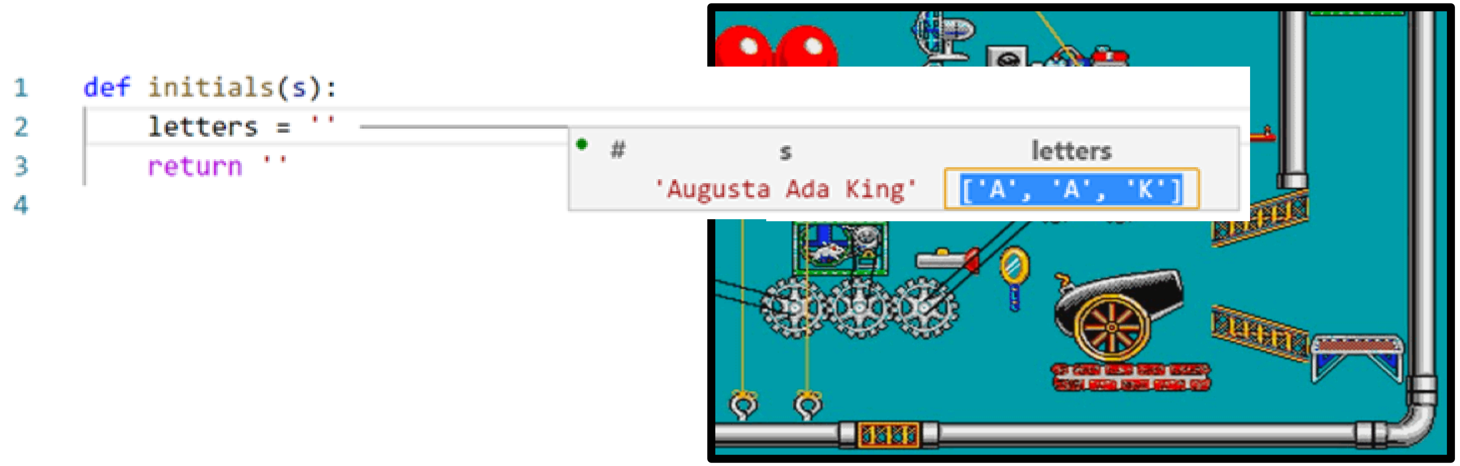


$\Downarrow$

p

Programming by Example

$$\text{input}_1 \rightarrow \text{output}_1$$
$$\text{input}_2 \rightarrow \text{output}_2$$
$$\text{input}_3 \rightarrow \text{output}_3$$
$$\Downarrow$$

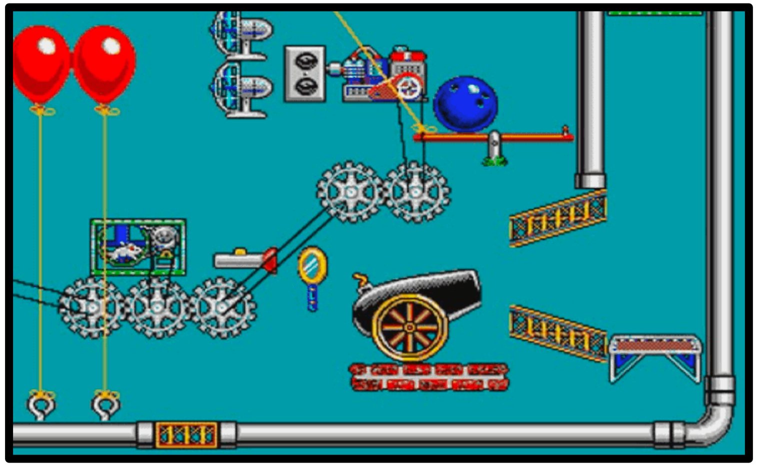


$$\Downarrow$$
$$p$$

Programming by Example

$\text{input}_1 \rightarrow \text{output}_1$
$\text{input}_2 \rightarrow \text{output}_2$
$\text{input}_3 \rightarrow \text{output}_3$

$\Downarrow$



$\Downarrow$

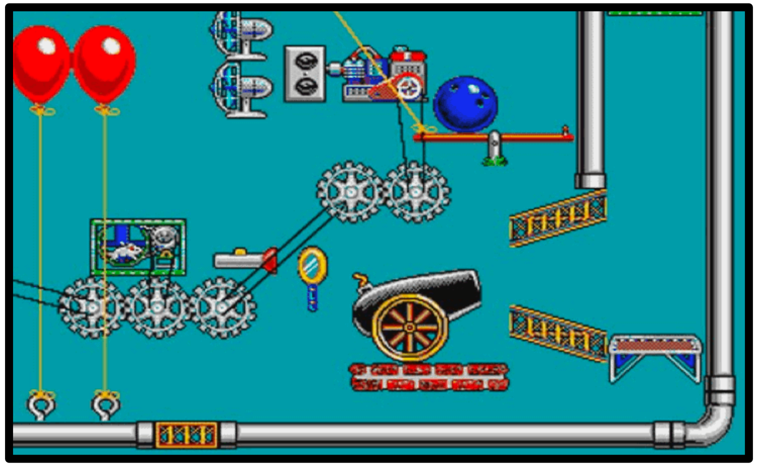
$$p$$

Counterexample
(input → output)

Programming by Example

Task: sum the min and max values in array of length n

⇓



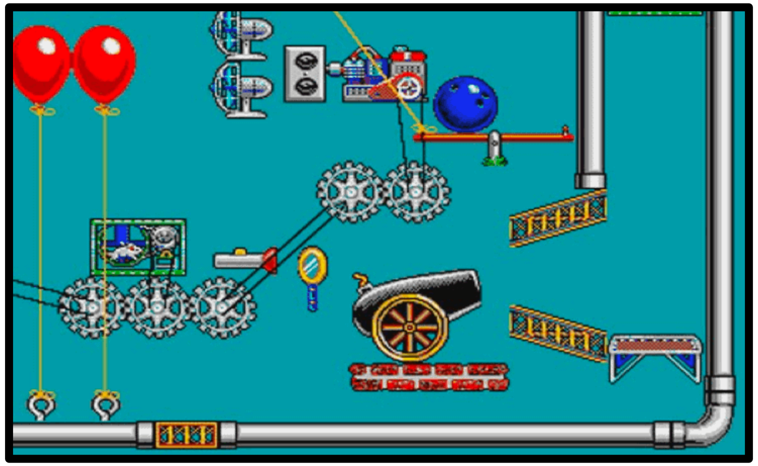
⇓

Programming by Example

Task: sum the min and max values in array of length n

{arr ↦[10,100,90,−1,2], n↦5} → 99

⇓



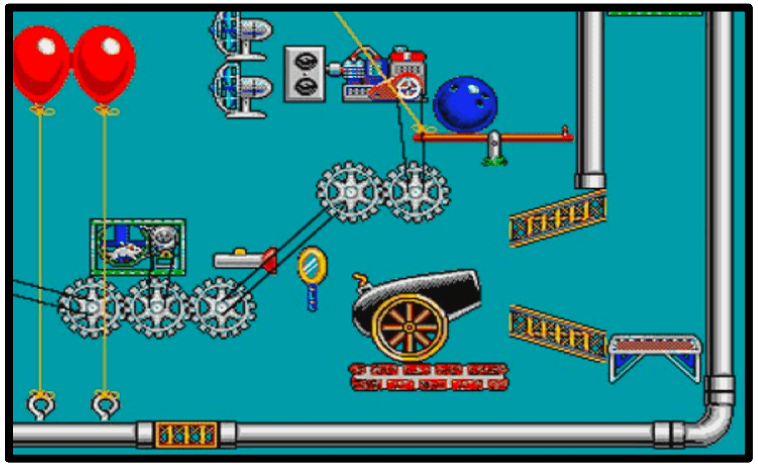
⇓

Programming by Example

Task: sum the min and max values in array of length n

$\{arr \mapsto [10,100,90,-1,2],\ n \mapsto 5\} \rightarrow 99$

$\Downarrow$



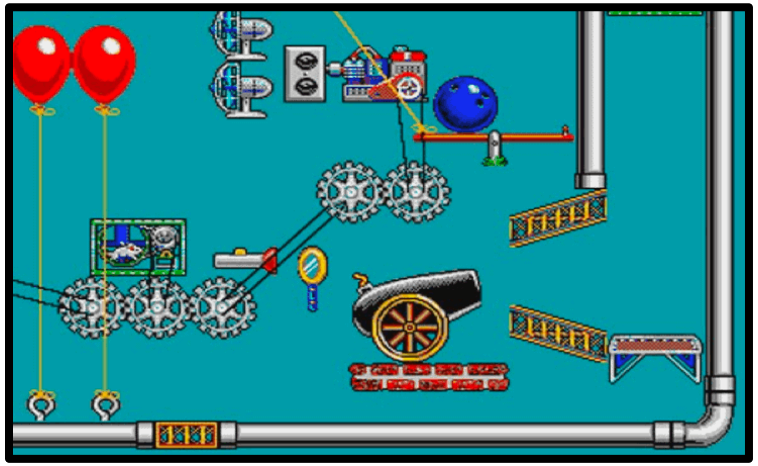
$\Downarrow$

Programming by Example

Task: sum the min and max values in array of length n

{arr ↦[10,100,90,−1,2], n↦5} → 99

⇓



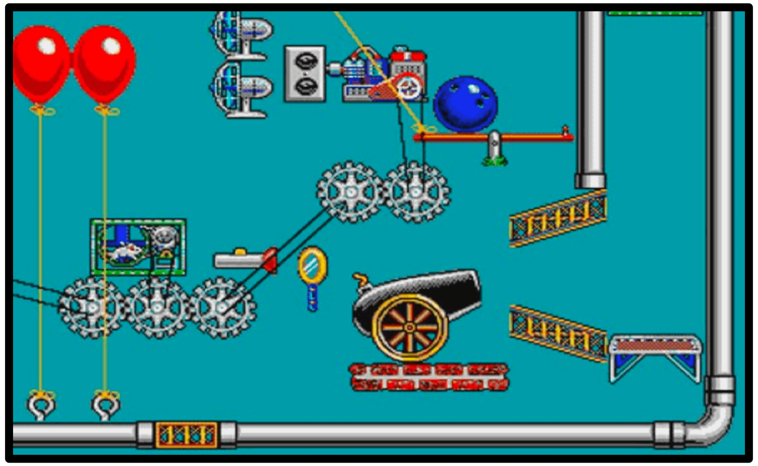
⇓

Programming by Example

Task: sum the min and max values in array of length n

$\{arr \mapsto [10, 100, 90, -1, 2], n \mapsto 5\} \rightarrow 99$

$\Downarrow$



$\Downarrow$

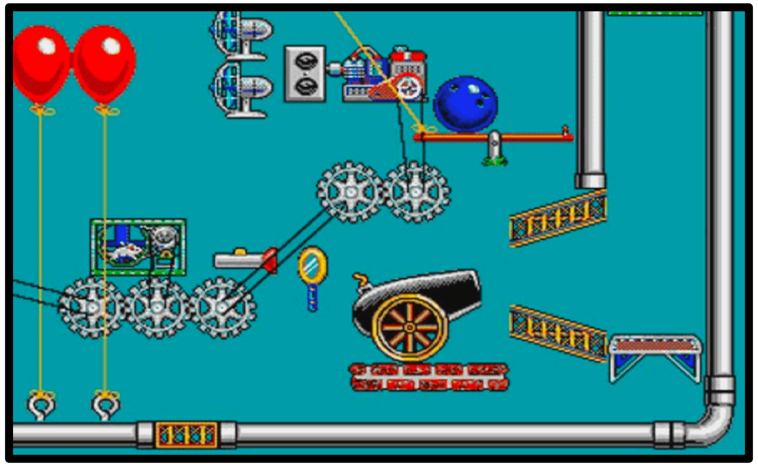
```
arr.sort()[0] + arr[n-1]
```

Programming by Example

Task: sum the min and max values in array of length n

{arr ↦[10,100,90,−1,2], n↦5} → 99

⇓



⇓

```
arr.sort()[0] + arr[n-1]
```

in-place 😱

Observational Equivalence

Enumerate space and test:

`n, arr, 0, 1, n + 0 n - 0, n + 1, n - 1, arr[0],…`

Observational Equivalence

Enumerate space and test:

`n, arr, 0, 1, n + 0 n - 0, n + 1, n - 1, arr[0],...`

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{\texttt{arr} \mapsto [10,100,90,-1,2],\ \texttt{n} \mapsto 5\}$

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{$arr ↦[10,100,90,−1,2], n↦5$\}$

n⟨5⟩

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

ι_1 = {arr ↦[10,100,90,−1,2], n↦5}

$n\langle 5 \rangle$

eval

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{\texttt{arr} \mapsto [10,100,90,-1,2],\ \texttt{n} \mapsto 5\}$

n⟨5⟩ arr⟨[10,100,90,−1,2]⟩

eval

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{$arr $\mapsto$[10,100,90,−1,2], n$\mapsto$5$\}$

n⟨5⟩ arr⟨[10,100,90,−1,2]⟩ 0⟨0⟩

eval

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

ι_1 = {arr ↦[10,100,90,−1,2], n↦5}

n⟨5⟩   arr⟨[10,100,90,−1,2]⟩   0⟨0⟩   1⟨1⟩

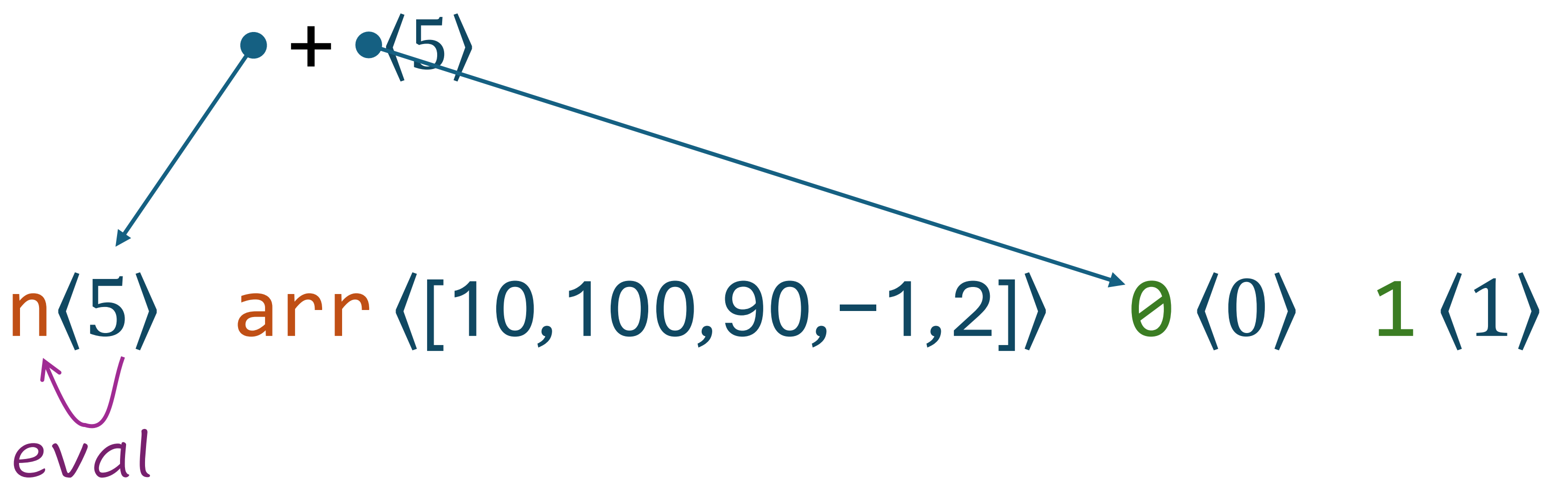
eval

Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{\texttt{arr} \mapsto [10,100,90,-1,2],\ \texttt{n} \mapsto 5\}$

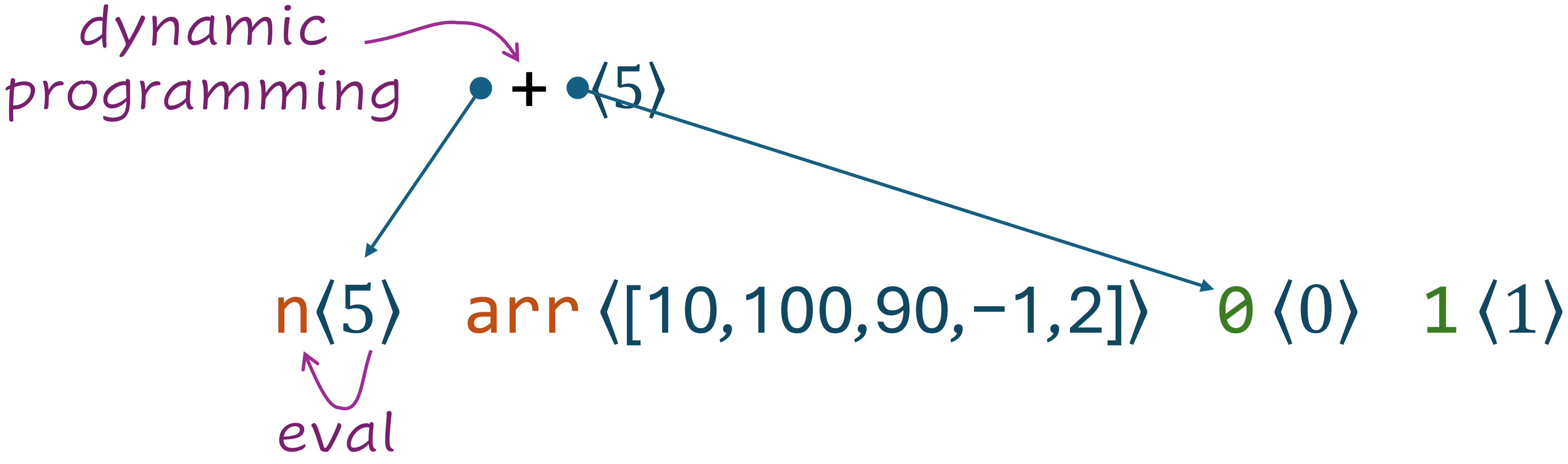


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{$arr $\mapsto$[10,100,90,−1,2], n$\mapsto$5$\}$

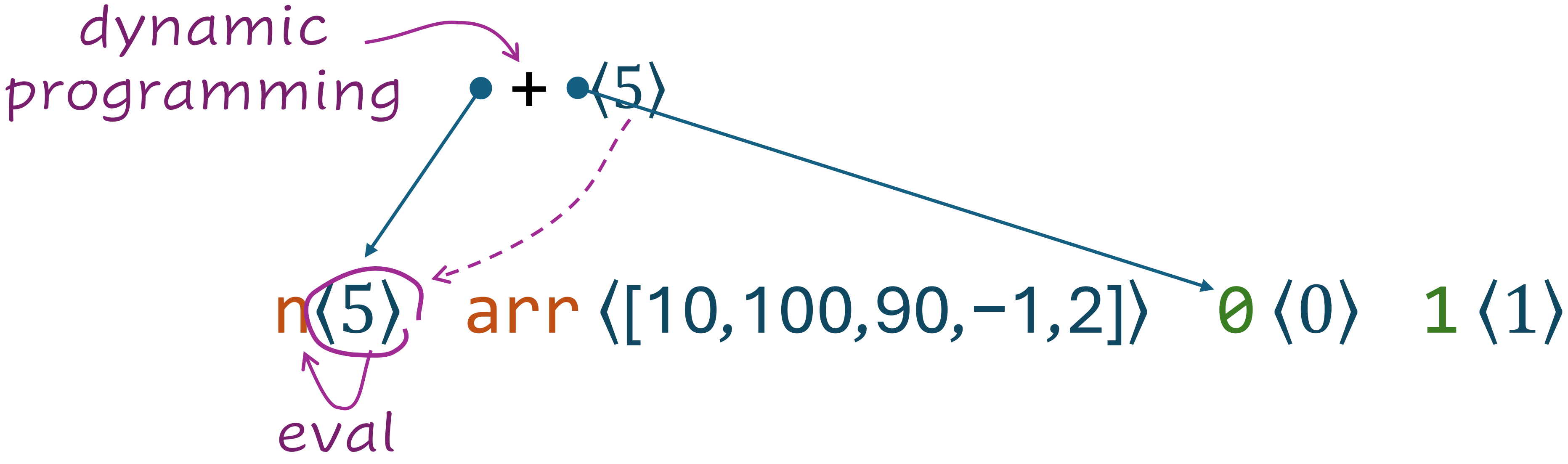


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

ι_1 = {arr ↦[10,100,90,−1,2], n↦5}

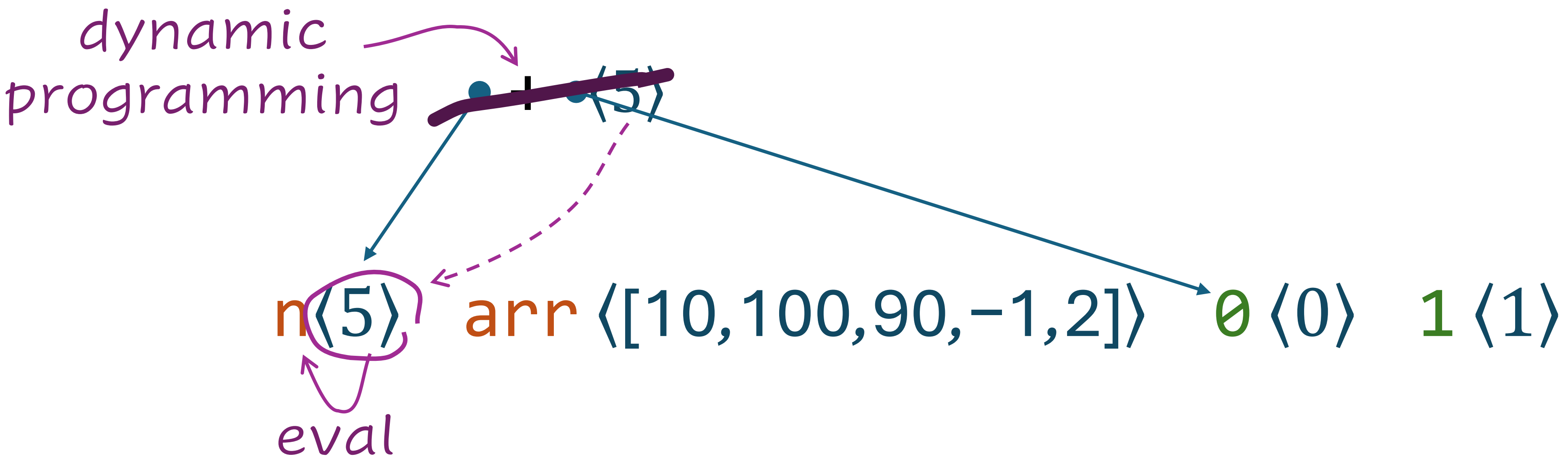


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{\texttt{arr} \mapsto [10,100,90,-1,2],\ \mathrm{n} \mapsto 5\}$

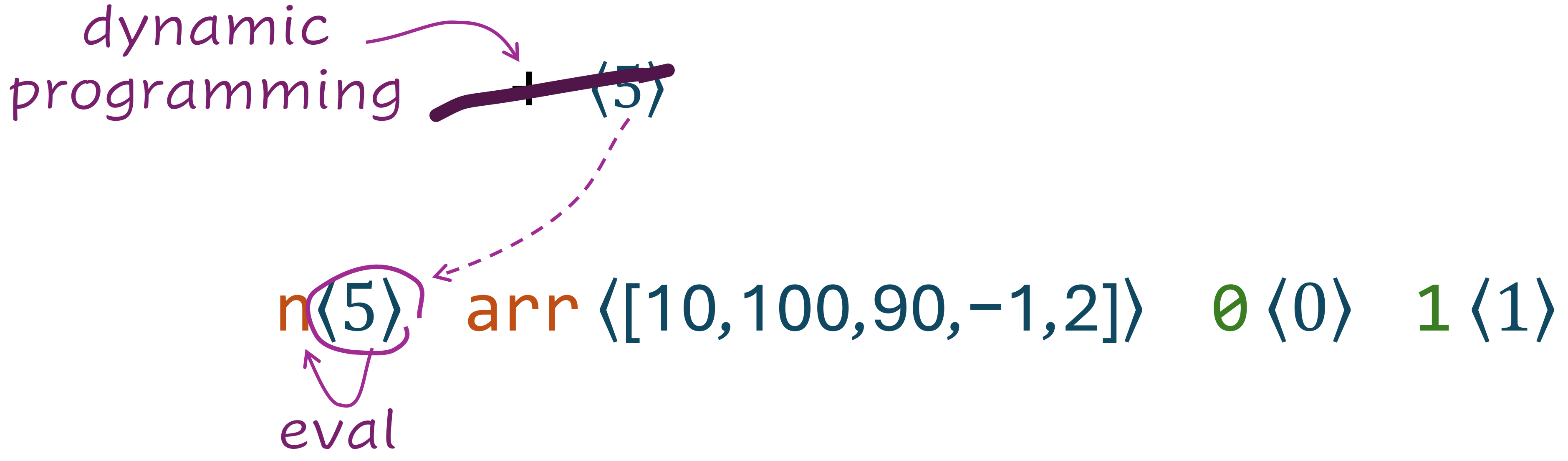


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{$arr $\mapsto$[10,100,90,−1,2], n$\mapsto$5$\}$

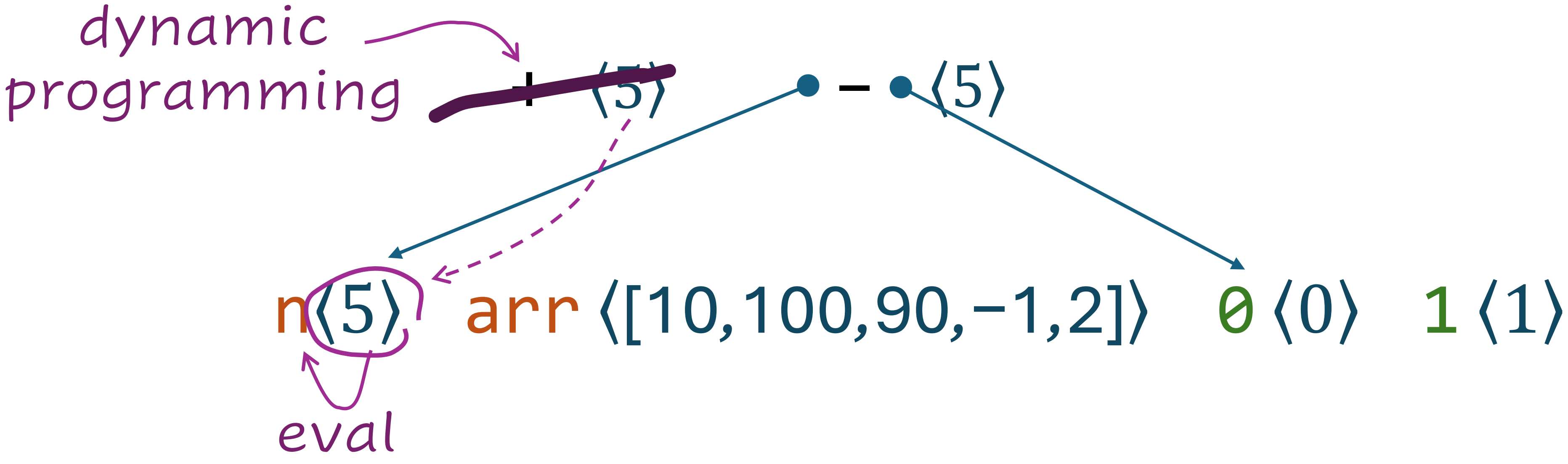


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{\texttt{arr} \mapsto [10,100,90,-1,2],\ \texttt{n} \mapsto 5\}$

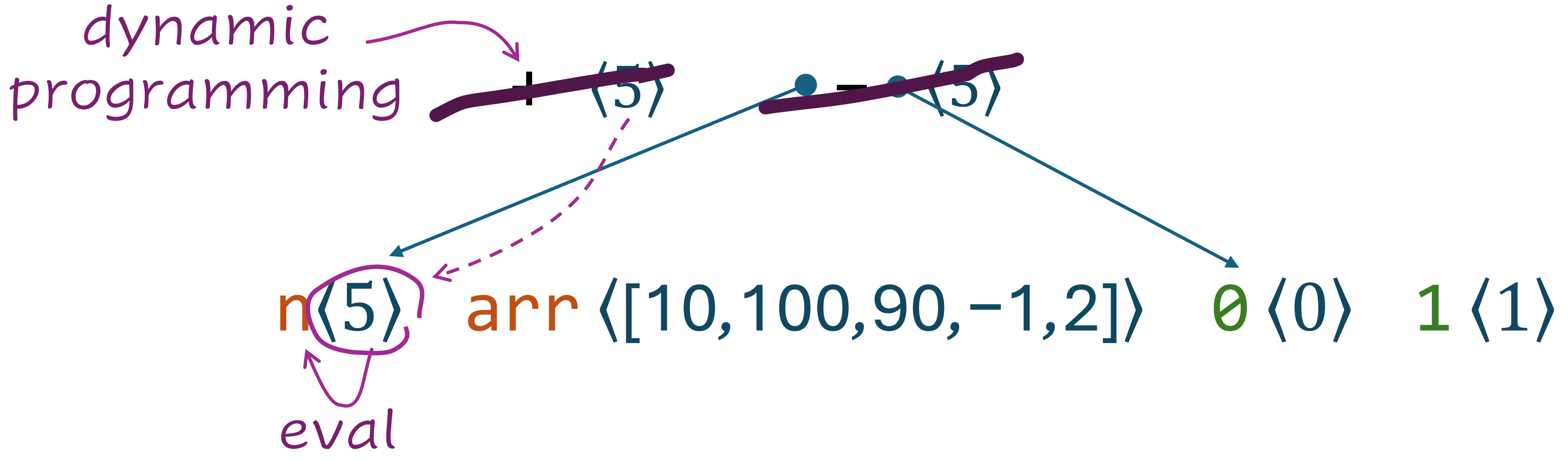


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{\texttt{arr} \mapsto [10,100,90,-1,2],\ \texttt{n} \mapsto 5\}$

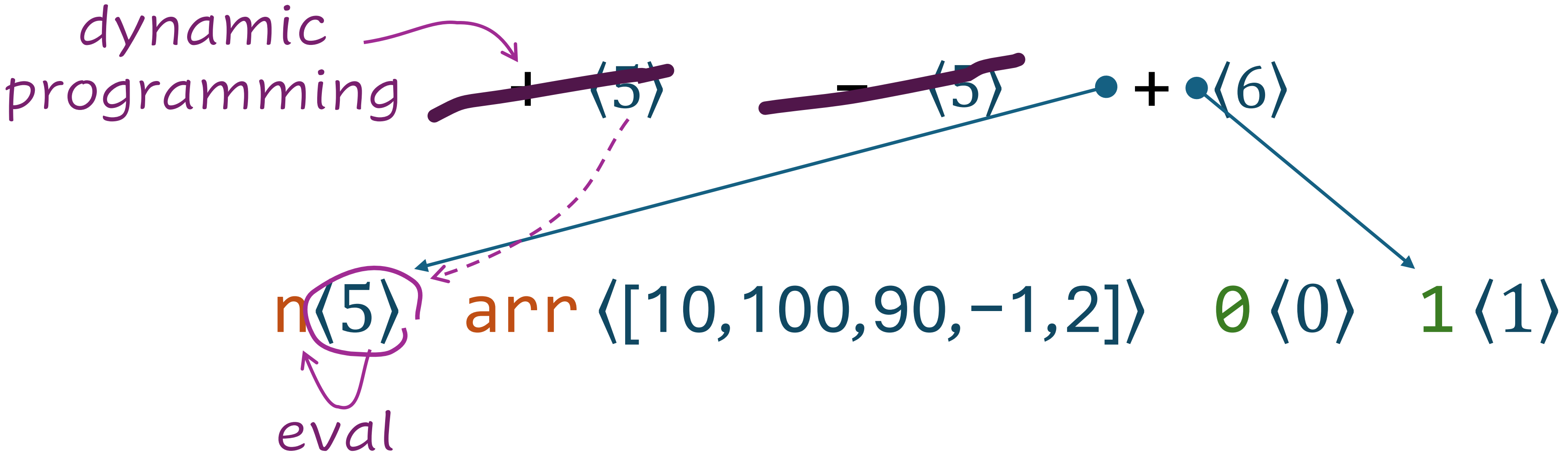


Observational Equivalence

ESCHER: Albarghouthi et al. 2013
TRANSIT: Udupa et al. 2013

Inputs that matter:

$\iota_1 = \{$arr $\mapsto$[10,100,90,−1,2], n$\mapsto$5$\}$



Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?

Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?

arr ⟨[10, 100, 90, −1, 2]⟩

Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?

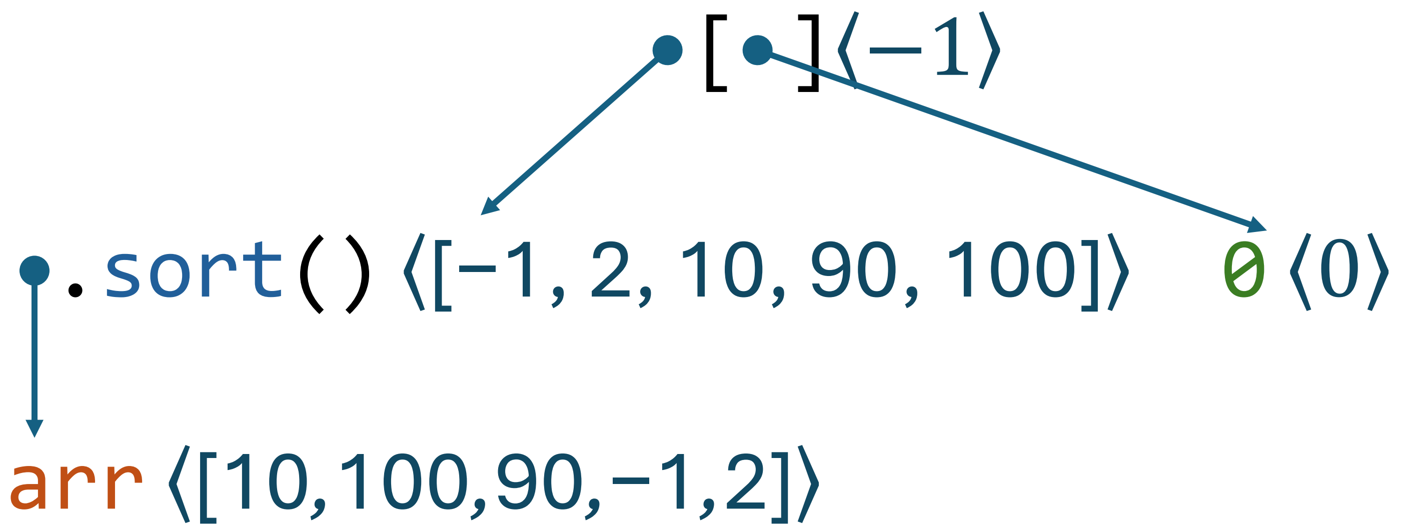
`.sort()` ⟨[−1, 2, 10, 90, 100]⟩

↓

`arr` ⟨[10,100,90,−1,2]⟩

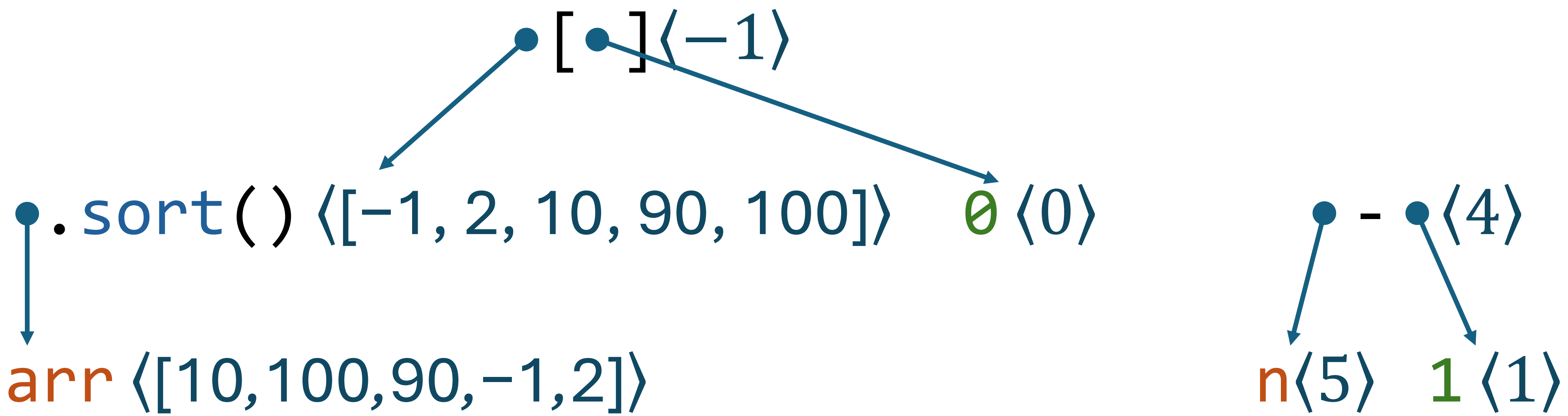
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



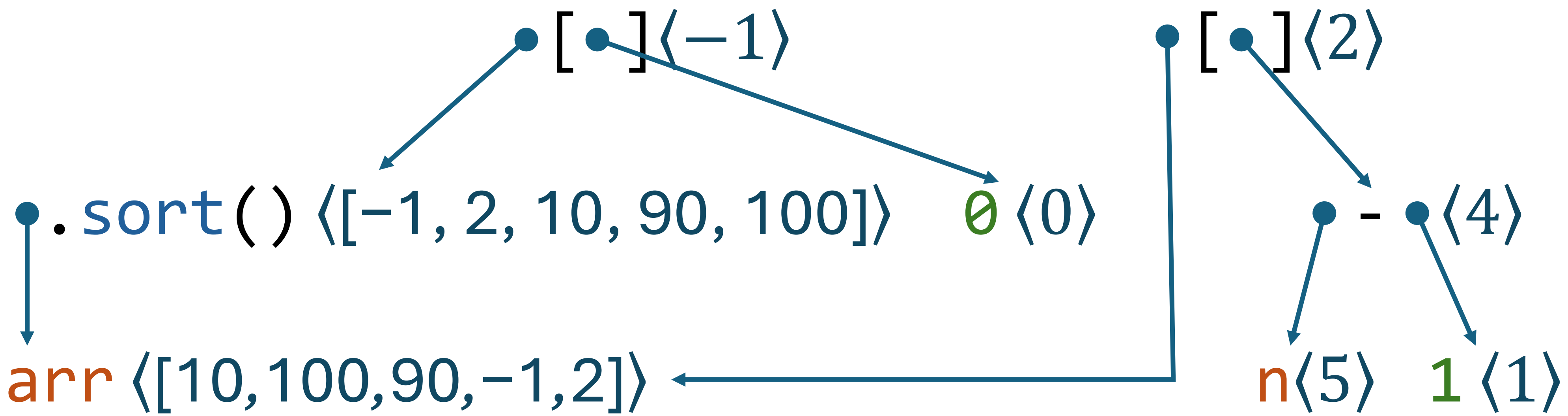
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



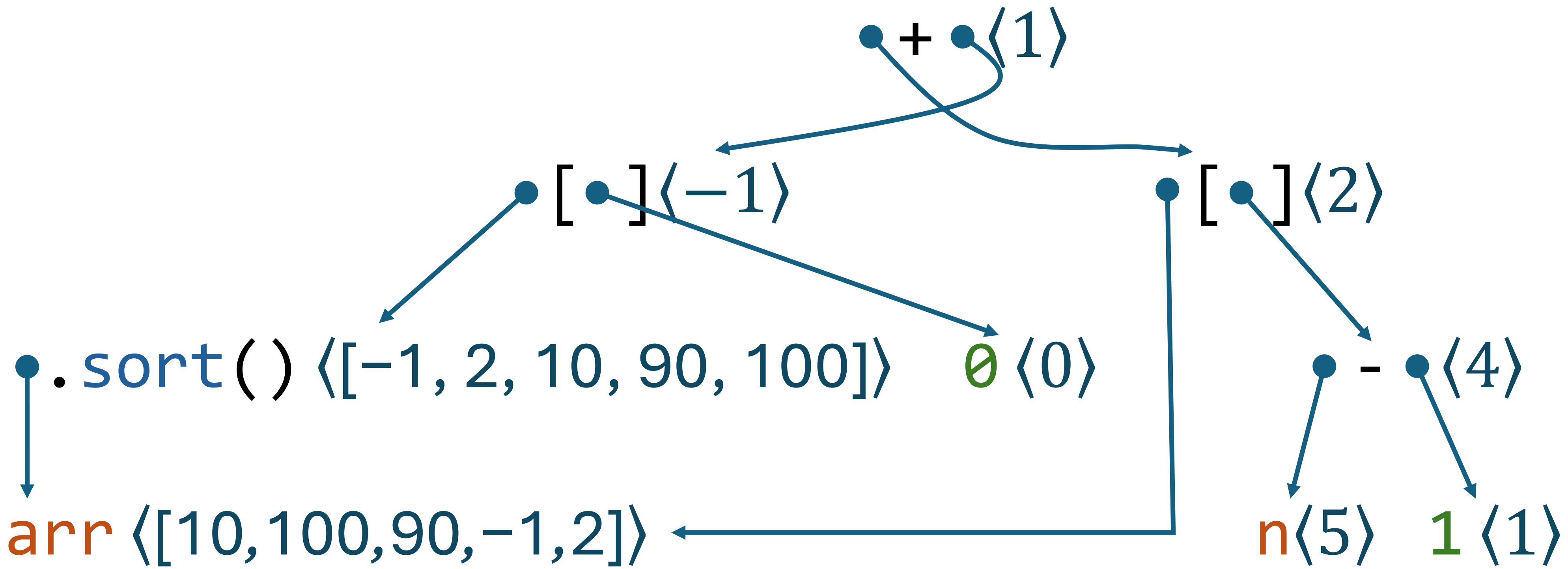
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



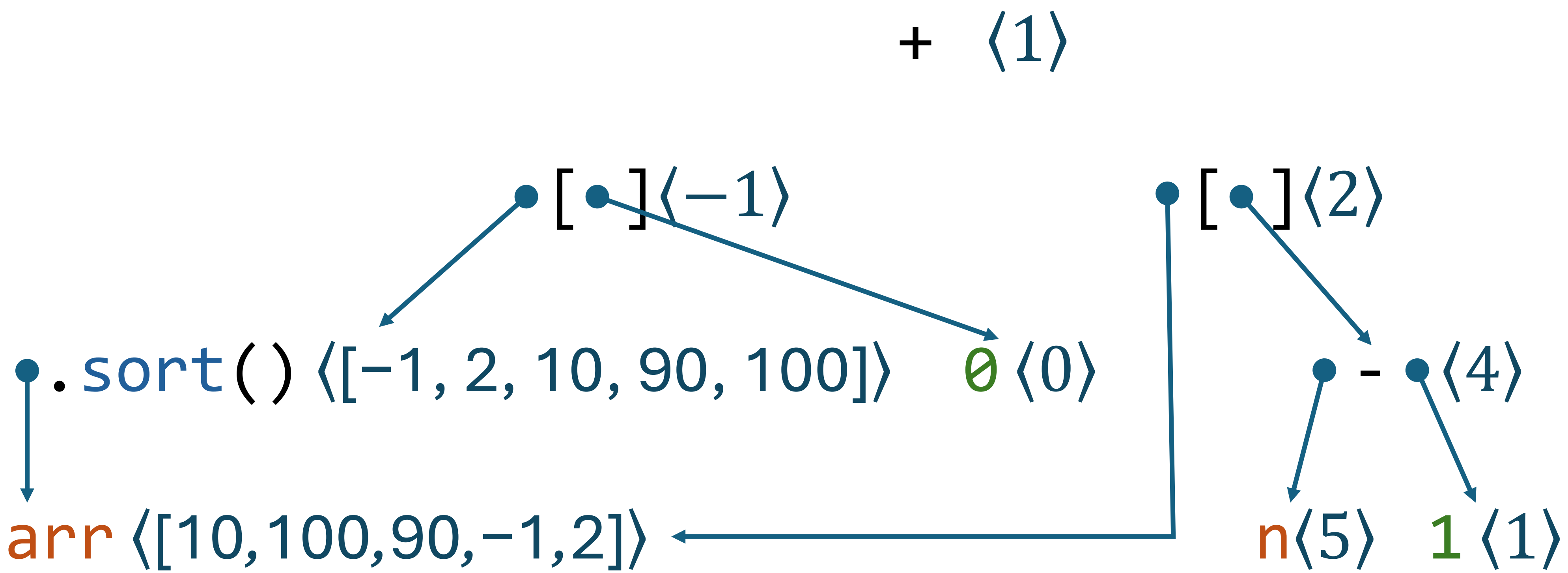
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



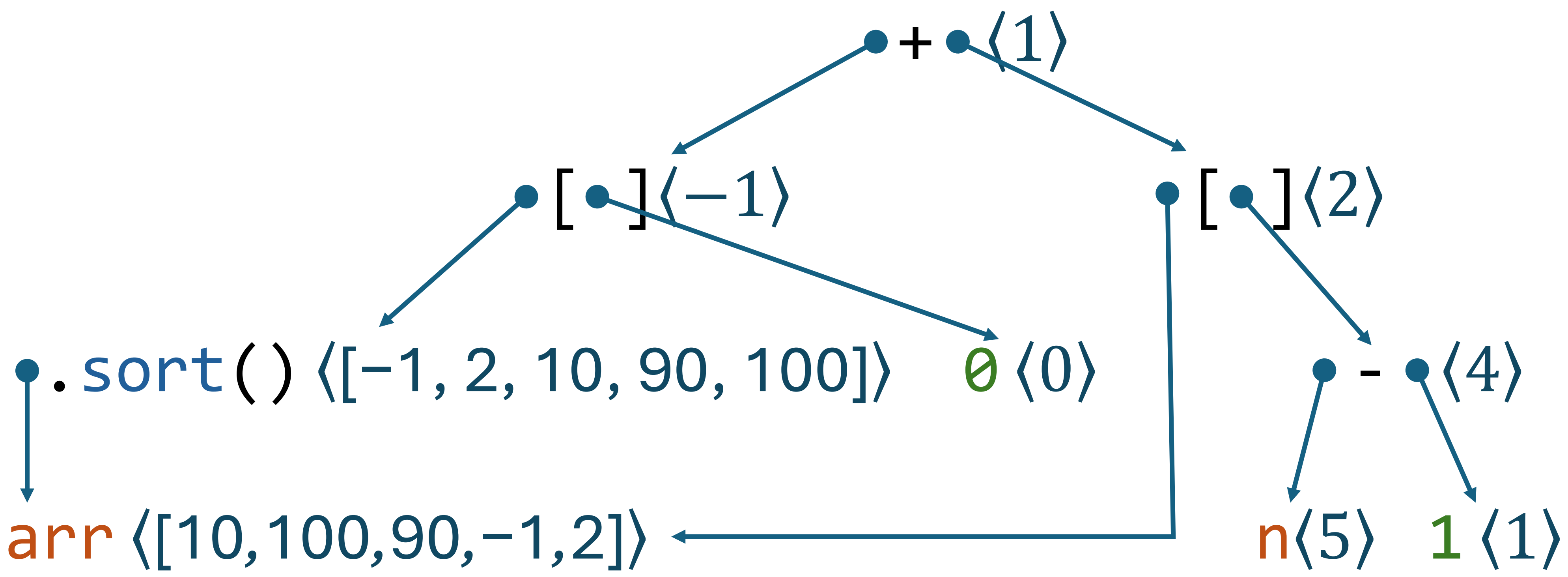
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



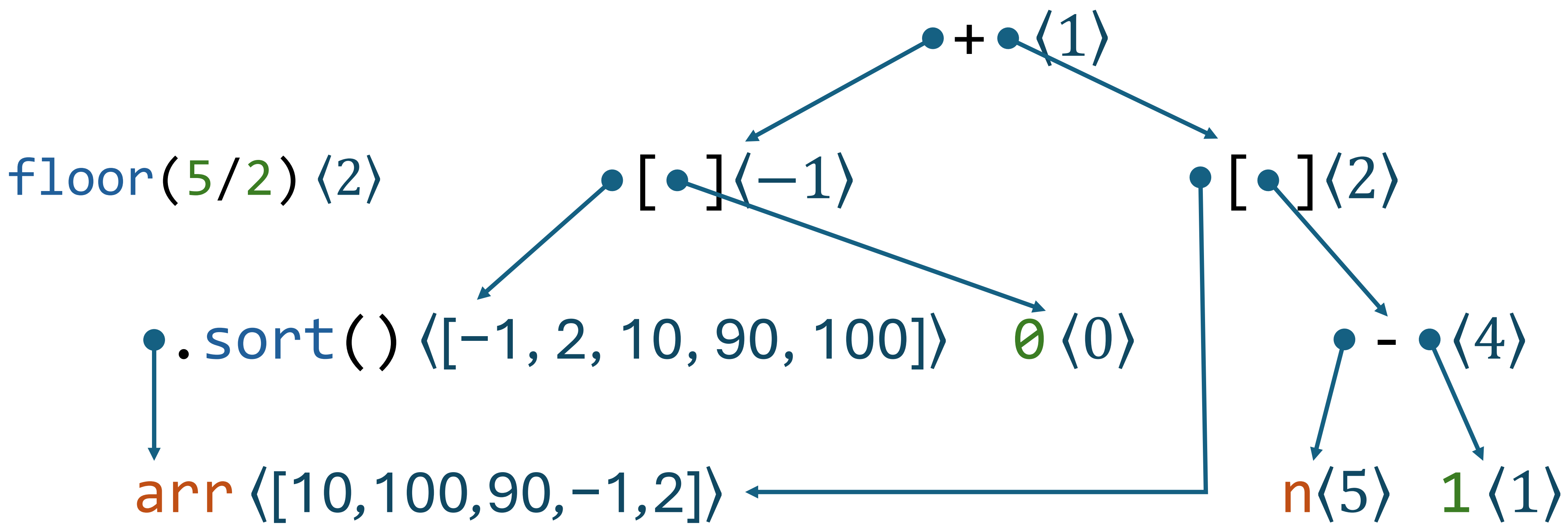
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



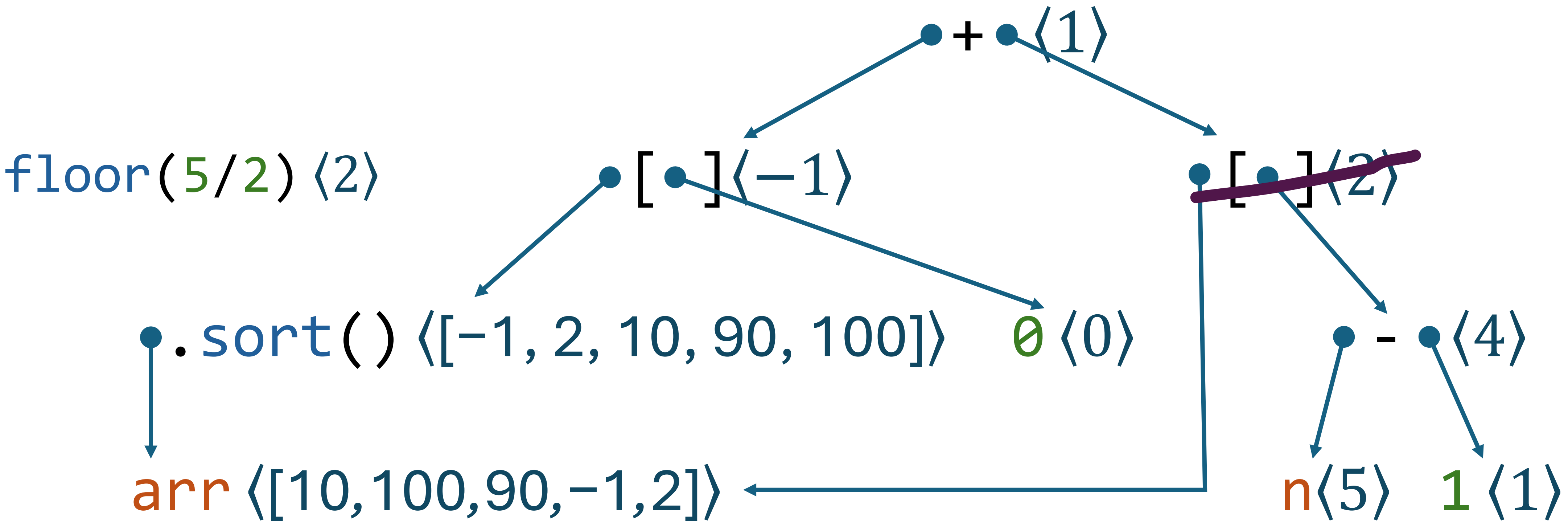
Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



Observational Equivalence and Mutations

Can we build `arr.sort()[0] + arr[n-1]`?



Let’s fix it!

Solution: ***S****ide-effects in* ***OB****servational* ***EQ****uivalence*

Let’s fix it!

Solution: ***S**ide-effects in **OB**servational **EQ**uivalence*

1) Add before- and after-states to representation

Let’s fix it!

Solution: ***S**ide-effects in **OB**servational **EQ**uivalence*

1) Add before- and after-states to representation
2) Trim states to bare necessities

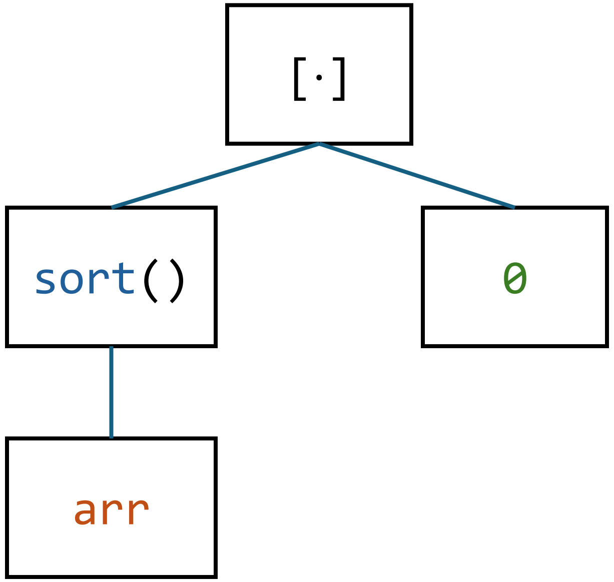
Let’s fix it!

Solution: ***S****ide-effects in* ***OB****servational* ***EQ****uivalence*

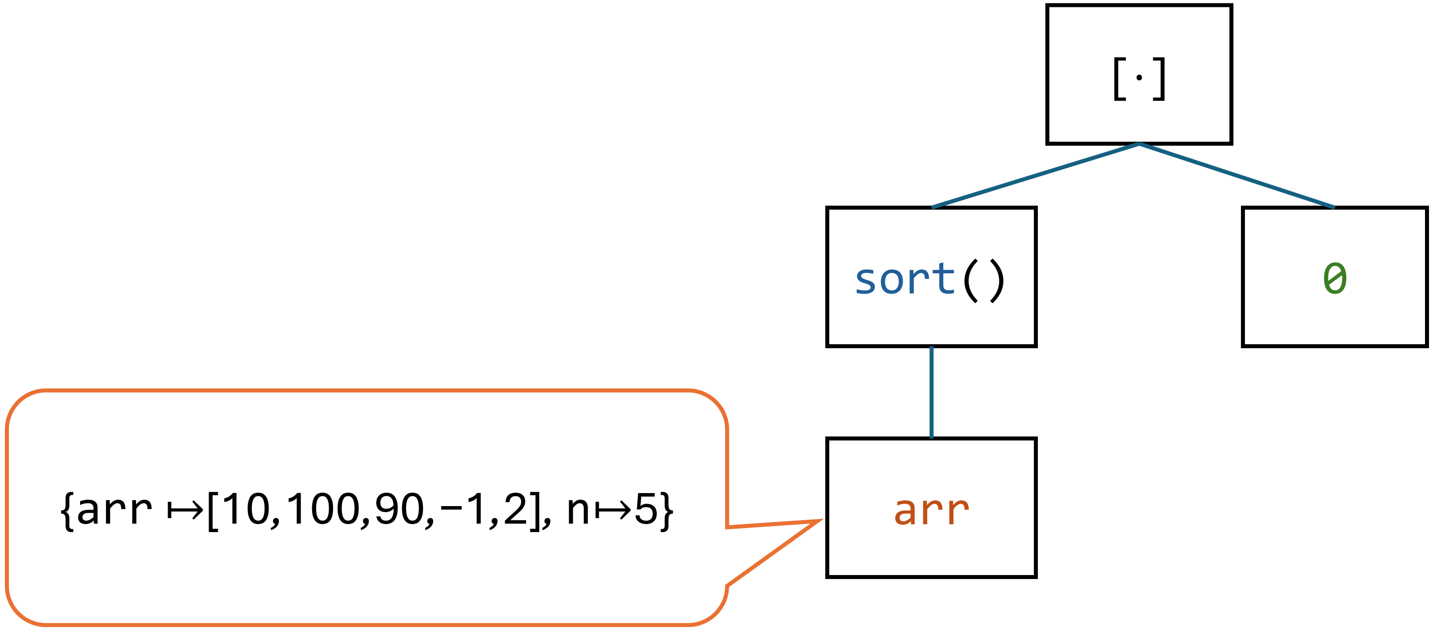
1) Add before- and after-states to representation
2) Trim states to bare necessities

Correct enumeration with mutations!

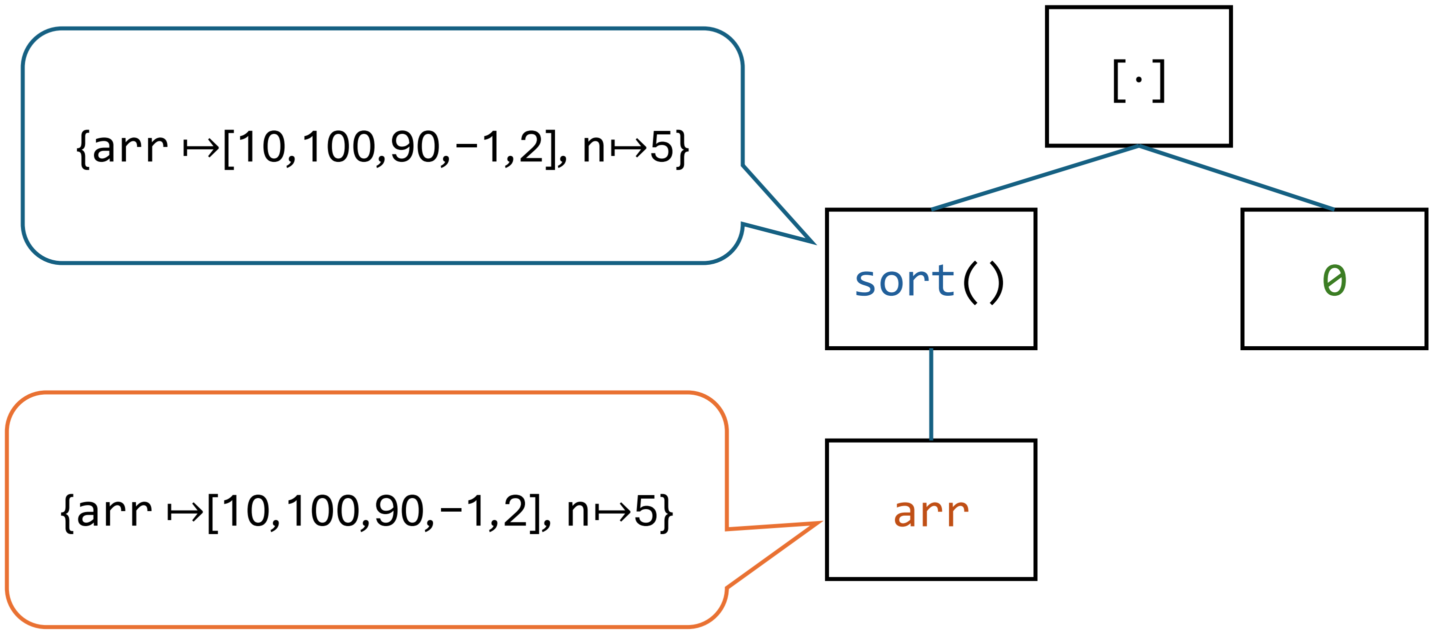
Function arguments and sequencing



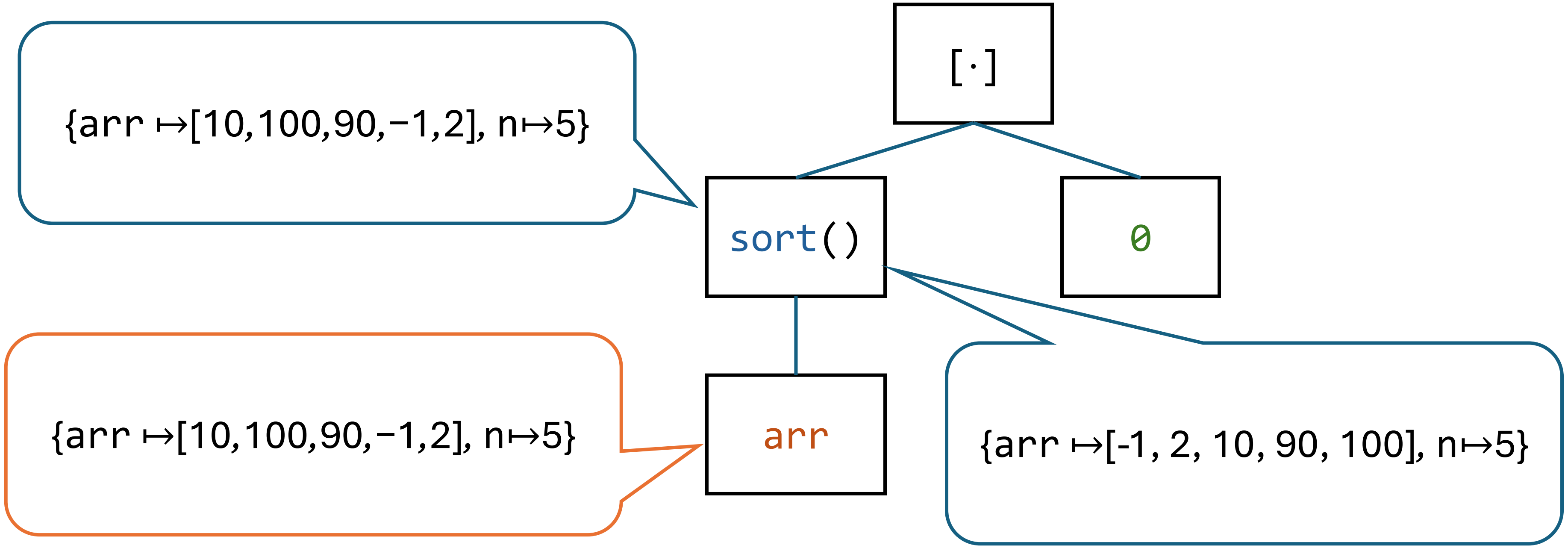
Function arguments and sequencing



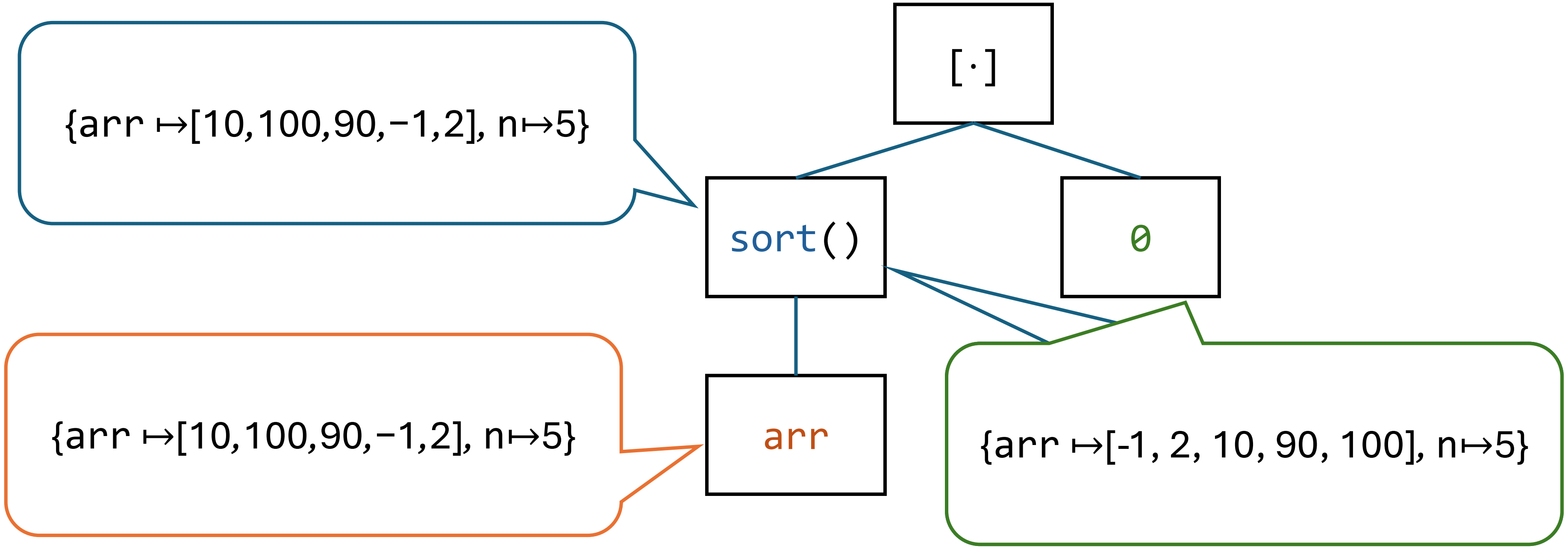
Function arguments and sequencing



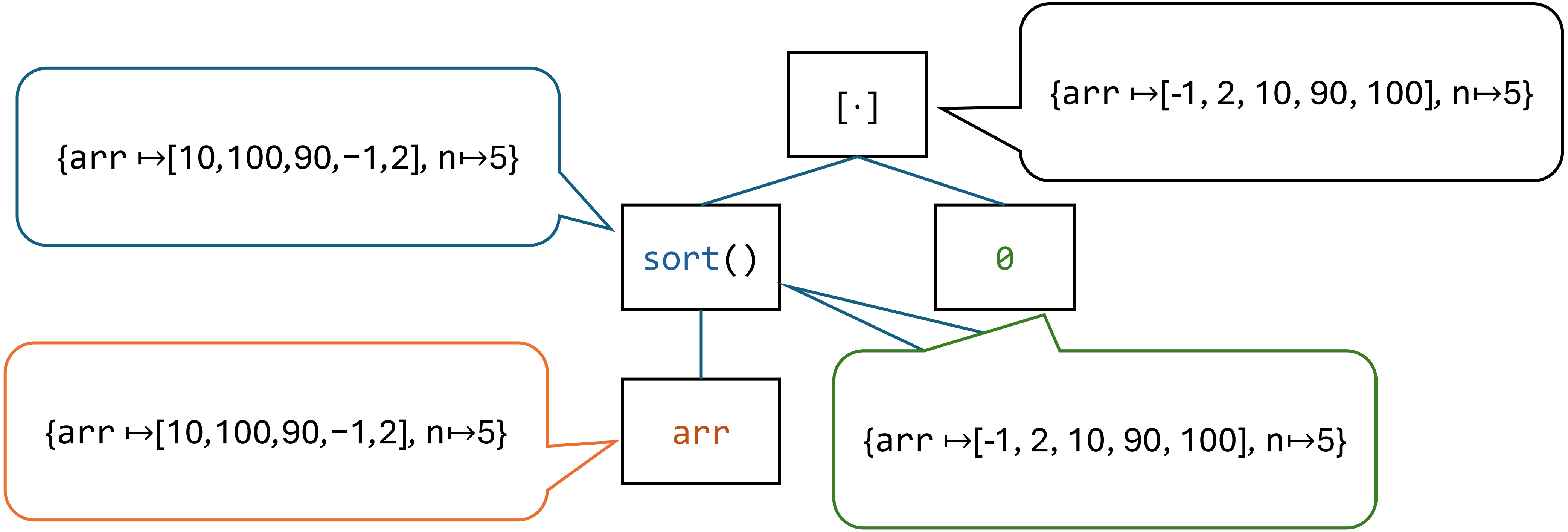
Function arguments and sequencing



Function arguments and sequencing



Function arguments and sequencing



Key idea 1: triples

$$\underbrace{\{x \mapsto v_x, y \mapsto v_y, \ldots\}}_{\text{precondition}}\ p\ \underbrace{\{x \mapsto v'_x, y \mapsto v'_y, \ldots; r\}}_{\text{postcondition}}$$

Key idea 1: triples

state after eval

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\; p \;\{x \mapsto v'_x, y \mapsto v'_y, \ldots; r\}$$

precondition

postcondition

Key idea 1: triples

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\ p\ \{x \mapsto v'_x, y \mapsto v'_y, \ldots; r\}$$

Key idea 1: triples

state after eval

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\ p\ \{x \mapsto v'_x, y \mapsto v'_y, \ldots; r\}$$

precondition

postcondition

result

{arr ↦[10,100,90,−1,2]} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100]; -1}

Key idea 1: triples

state after eval

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\ p\ \{x \mapsto v'_x, y \mapsto v'_y, \ldots; r\}$$

precondition

postcondition

result

{arr ↦[10,100,90,−1,2]} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100]; -1}

side effect!

Key idea 1: triples

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\; p\; \{x \mapsto v'_x, y \mapsto v'_y, \ldots; r\}$$

valid sequence

$$\{x \mapsto {v'}_x, y \mapsto {v'}_y, \ldots\}\; p'\; \{x \mapsto v''_x, y \mapsto v''_y, \ldots; r'\}$$

Key idea 1: triples

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\ p;$$

$$p'\ \{x \mapsto v_x'', y \mapsto v_y'', \ldots; r'\}$$

Key idea 1: triples

initial state — *arr is sorted*

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

{arr ↦[-1, 2, 10, 90, 100], n↦5}`arr[n-1]` {arr ↦[-1, 2, 10, 90, 100], n↦5;100}

arr is sorted

Key idea 1: triples

initial state

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

arr is sorted

valid sequence

{arr ↦[-1, 2, 10, 90, 100], n↦5}`arr[n-1]` {arr ↦[-1, 2, 10, 90, 100], n↦5;100}

arr is sorted

Key idea 1: triples

initial state

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

arr is sorted

valid sequence

{arr ↦[-1, 2, 10, 90, 100], n↦5}`arr[n-1]` {arr ↦[-1, 2, 10, 90, 100], n↦5;100}

arr is sorted

✓ Apply `f(?,?)`

Key idea 1: triples

initial state

arr is sorted

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

<u>**in**</u>valid sequence!

{arr ↦[10,100,90,−1,2], n↦5} `arr[n-1]` {arr ↦[10,100,90,−1,2], n↦5;2}

initial state

initial state

Key idea 1: triples

initial state — *arr is sorted*

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

invalid sequence!

{arr ↦[10,100,90,−1,2], n↦5} `arr[n-1]` {arr ↦[10,100,90,−1,2], n↦5;2}

initial state — *initial state*

✘ Apply `f(?,?)`

Key idea 1: triples

{arr ↦[10,100,90,−1,2], n↦5} `arr[n-1]` {arr ↦[10,100,90,−1,2], n↦5;2}

valid sequence

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

Key idea 1: triples

{arr ↦[10,100,90,−1,2], n↦5} `arr[n-1]` {arr ↦[10,100,90,−1,2], n↦5;2}

valid sequence

{arr ↦[10,100,90,−1,2], n↦5} `arr.sort()[0]` {arr ↦[-1, 2, 10, 90, 100], n↦5;-1}

✓ Apply `f(?,?)`

Key idea 1: triples

$$p\langle r\rangle$$

Key idea 1: triples

$$\langle r \rangle$$

Key idea 1: triples

$$\langle r \rangle$$

$$\{initial\ state\}\ p\ \{initial\ state; r\}$$

Key idea 1: triples

$$\cancel{p}\langle r \rangle$$

$$\{initial\ state\}\ p\ \{initial\ state; r\}$$

$$\{x \mapsto v_x, y \mapsto v_y, \dots\}\ p\ \{x \mapsto v_x', y \mapsto v_y', \dots; r\}$$

Key idea 1: triples

$$\langle r\rangle$$

$$\{initial\ state\}\ p\ \{initial\ state; r\}$$

$$\{x \mapsto v_x, y \mapsto v_y, \ldots\}\ \{x \mapsto v_x', y \mapsto v_y', \ldots; r\}$$

Is this stupid?

initial state

{arr ↦[10,100,90,−1,2], n↦5} 0 {arr ↦[10,100,90,−1,2], n↦5; 0}

Is this stupid?

initial state

{arr ↦[10,100,90,−1,2], n↦5} 0 {arr ↦[10,100,90,−1,2], n↦5; 0}

arr is sorted

{arr ↦[-1, 2, 10, 90, 100], n↦5} 0 {arr ↦[-1, 2, 10, 90, 100], n↦5; 0}

Is this stupid?

initial state

{arr ↦[10,100,90,−1,2], n↦5} `0` {arr ↦[10,100,90,−1,2], n↦5; 0}

arr is sorted

{arr ↦[-1, 2, 10, 90, 100], n↦5} `0` {arr ↦[-1, 2, 10, 90, 100], n↦5; 0}

arr is empty

{arr ↦[], n↦5} `0` {arr ↦[], n↦5; 0}

Is this stupid?

initial state

{arr ↦ [10,100,90,−1,2], n ↦ 5} `0` {arr ↦ [10,100,90,−1,2], n ↦ 5; 0}

arr is sorted

{arr ↦ [-1, 2, 10, 90, 100], n ↦ 5} `0` {arr ↦ [-1, 2, 10, 90, 100], n ↦ 5; 0}

arr is empty

{arr ↦ [], n ↦ 5} `0` {arr ↦ [], n ↦ 5; 0}

{arr ↦ [10,100,90,−1], n ↦ 5} `0` {arr ↦ [10,100,90,−1], n ↦ 5; 0}

Separation Logic!

one part of the heap * another part of the heap

Separation Logic!

one part of the heap $*$ another part of the heap

$$\frac{\{P\}\ p\ \{Q\}}{\{P * R\}\ p\ \{Q * R\}}\ \textsc{Frame}$$

Key idea 2: local triplets using SL

$$\underbrace{\{x \mapsto v_x\ ,\ \ y \mapsto v_y, \ldots\}}_{\text{precondition}}\ p\ \{\underbrace{x \mapsto v'_x\ ,\ y \mapsto v'_y, \ldots ;\ \underbrace{r}_{\text{result}}}_{\text{postcondition}}\}$$

Key idea 2: local triplets using SL

$$\underbrace{\{x \mapsto v_x * \; y \mapsto v_y, \ldots\}}_{\text{precondition}} \; p \; \underbrace{\{x \mapsto v'_x * y \mapsto v'_y, \ldots; r\}}_{\text{postcondition}}$$

result

Key idea 2: local triplets using SL

$$\{x \mapsto v_x * y \mapsto v_y, \ldots\}\; p \;\{x \mapsto v'_x * y \mapsto v'_y, \ldots ; r\}$$

precondition

postcondition

result

$\{arr \mapsto [-1, 2, 10, 90, 100] * n \mapsto 5\}$ `n - 1` $\{arr \mapsto [-1, 2, 10, 90, 100] * n \mapsto 5; 4\}$

Key idea 2: local triplets using SL

$$\{x \mapsto v_x * \ y \mapsto v_y, \ldots\}\ p\ \{x \mapsto v_x' * y \mapsto v_y', \ldots; r\}$$

precondition

postcondition

result

{~~arr ↦ [-1, 2, 10, 90, 100]~~ * n↦5} n - 1 {~~arr ↦ [-1, 2, 10, 90, 100]~~ * n↦5; 4}

Key idea 2: local triplets using SL

$$\{\underbrace{x \mapsto v_x * y \mapsto v_y, \dots}_{\text{precondition}}\}\ p\ \{\underbrace{x \mapsto v'_x * y \mapsto v'_y, \dots;}_{\text{postcondition}}\ \underset{\uparrow\ \text{result}}{r}\}$$

{~~arr ↦[-1, 2, 10, 90, 100]~~ * n↦5} `n - 1` {~~arr ↦[-1, 2, 10, 90, 100]~~ * n↦5; 4}

{arr ↦[-1, 2, 10, 90, 100] * n↦5} `0` {arr ↦[-1, 2, 10, 90, 100] * n↦5; 0}

Key idea 2: local triplets using SL

$$\{x \mapsto v_x * y \mapsto v_y, \ldots\}\ p\ \{x \mapsto v_x' * y \mapsto v_y', \ldots; r\}$$

precondition

postcondition

result

{~~arr ↦ [-1, 2, 10, 90, 100]~~ * n↦5} n – 1 {~~arr ↦ [-1, 2, 10, 90, 100]~~ * n↦5; 4}

{~~arr ↦ [-1, 2, 10, 90, 100] * n↦~~5} 0 {~~arr ↦ [-1, 2, 10, 90, 100] * n↦~~5; 0}

Key idea 2: local triplets using SL

$$\{x \mapsto v_x * \ y \mapsto v_y, \dots\}\ p\ \{x \mapsto v'_x * y \mapsto v'_y, \dots ; r\}$$

precondition

postcondition

result

{arr ↦ [-1, 2, 10, 90, 100] * n↦5} n - 1 {arr ↦ [-1, 2, 10, 90, 100] * n↦5; 4}

{arr ↦ [-1, 2, 10, 90, 100] {emp} 0 {emp; 0} 1, 2, 10, 90, 100] * n↦5; 0}

Separation Logic!

one part of the heap $*$ another part of the heap

$$\frac{\{P\}\ p\ \{Q\}}{\{P * R\}\ p\ \{Q * R\}}\ \textsc{Frame}$$

Key idea 2: local triplets using SL

$$\frac{\{P\}\ p\ \{Q; r\}}{\{P * R\}\ p\ \{Q * R; r\}}\ \textsc{Frame}$$

Key idea 2: local triplets using SL

$$\frac{\{P\}\ p\ \{Q; r\}}{\{P * R\}\ p\ \{Q * R; r\}}\ \textsc{Frame}$$

$$\frac{\{P_1\}\ p_1\ \{P_2; r_1\}\ \{P_2\}\ p_2\ \{P_3; r_2\} \cdots \{P_k\}\ p_k\ \{P_{k+1}; r_k\} \quad (c(r_1, \ldots, r_k), P_{k+1}) \rightarrow (r, Q)}{\{P_1\}\ c(p_1, \ldots, p_k)\ \{Q; r\}}\ \textsc{Eval}$$

Key idea 2: local triplets using SL

$$\frac{\{P\}\ p\ \{Q; r\}}{\{P * R\}\ p\ \{Q * R; r\}}\ \textsc{Frame}$$

$$\frac{\{P_1\}\ p_1\ \{P_2; r_1\}\ \{P_2\}\ p_2\ \{P_3; r_2\} \cdots \{P_k\}\ p_k\ \{P_{k+1}; r_k\} \quad (c(r_1, \ldots, r_k), P_{k+1}) \to (r, Q)}{\{P_1\}\ c(p_1, \ldots, p_k)\ \{Q; r\}}\ \textsc{Eval}$$

interpreter

Program proofs with EVAL and FRAME

[-1, 2, 10, 90, 100]

{arr ↦ v_{arr}} arr {arr ↦ v_{arr}; v_{arr}}

{emp} 1 {emp; 1}

{n↦ 5} n {n↦ 5; 5}

Program proofs with Eval and Frame

[-1, 2, 10, 90, 100]

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\, v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\}$

$$\frac{\{emp\}\ \texttt{1}\ \{emp;\, 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5;\, 1\}}\ \textsc{Frame}$$

Program proofs with EVAL and FRAME

[-1, 2, 10, 90, 100]

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\ v_{arr}\}$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\ 5\} \qquad \dfrac{\{emp\}\ \texttt{1}\ \{emp;\ 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5;\ 1\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5;\ 4\}}\ \textsc{Eval}$$

Program proofs with Eval and Frame

[-1, 2, 10, 90, 100]

$$\frac{\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\, v_{arr}\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\, v_{arr}\}}\ \textsc{Frame}$$

$$\frac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\} \qquad \dfrac{\{emp\}\ \texttt{1}\ \{emp;\, 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5;\, 1\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5;\, 4\}}\ \textsc{Eval}$$

Program proofs with EVAL and FRAME

[-1, 2, 10, 90, 100]

$$\frac{\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\, v_{arr}\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\, v_{arr}\}}\ \textsc{Frame}$$

$$\frac{\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\} \qquad \dfrac{\{emp\}\ \texttt{1}\ \{emp;\, 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5;\, 1\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5;\, 4\}}\ \textsc{Eval}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\, 4\}}\ \textsc{Frame}$$

Program proofs with EVAL and FRAME

[-1, 2, 10, 90, 100]

$$\dfrac{\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5; v_{arr}\}}\ \textsc{Frame}$$

$$\dfrac{\{emp\}\ \texttt{1}\ \{emp; 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5; 1\}}\ \textsc{Frame}$$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\} \qquad \{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5; 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5; 4\}}\ \textsc{Eval}$$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5; 4\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5; 4\}}\ \textsc{Frame}$$

$$\dfrac{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5; v_{arr}\} \qquad \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5; 4\}}{\quad}\ \textsc{Eval}$$

Program proofs with EVAL and FRAME

[-1, 2, 10, 90, 100]

$$\dfrac{\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\ v_{arr}\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\ v_{arr}\}}\ \textsc{Frame}$$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\ 5\} \qquad \dfrac{\{emp\}\ \texttt{1}\ \{emp;\ 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5;\ 1\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5;\ 4\}}\ \textsc{Eval}$$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{n} \mapsto 5;\ 4\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\ 4\}}\ \textsc{Frame}$$

$$\dfrac{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr}\ \{\ldots;\ v_{arr}\} \qquad \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{n - 1}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\ 4\}}{\{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5\}\ \texttt{arr[n - 1]}\ \{\texttt{arr} \mapsto v_{arr} * \texttt{n} \mapsto 5;\ 100\}}\ \textsc{Eval}$$

Bottom-up proof search

$\{\mathtt{arr} \mapsto v_{arr}\}$ `arr` $\{\mathtt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\mathtt{n} \mapsto 5\}$ `n` $\{\mathtt{n} \mapsto 5; 5\}$

$\{emp\}$ `1` $\{emp; 1\}$

$\{emp\}$ `0` $\{emp; 0\}$

Bottom-up proof search

apply **+**

$\{\texttt{arr} \mapsto v_{arr}\}$ `arr` $\{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}$ `n` $\{\texttt{n} \mapsto 5; 5\}$

$\{emp\}$ `1` $\{emp; 1\}$

$\{emp\}$ `0` $\{emp; 0\}$

Bottom-up proof search

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1 2

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

apply **+**

Bottom-up proof search

$\{\texttt{arr} \mapsto v_{arr}\}$ arr $\{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}$ n $\{\texttt{n} \mapsto 5; 5\}$ 1 2

$\{emp\}$ 1 $\{emp; 1\}$

$\{emp\}$ 0 $\{emp; 0\}$

apply **+**

$\{\texttt{n} \mapsto 5\}$ n $\{\texttt{n} \mapsto 5; 5\}$ $\{\texttt{n} \mapsto 5\}$ n $\{\texttt{n} \mapsto 5; 5\}$

Bottom-up proof search

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1 2

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

apply **+**

$$\frac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\} \quad \{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}}{\{\texttt{n} \mapsto 5\}\ \texttt{n} + \texttt{n}\ \{\texttt{n} \mapsto 5; 10\}}\ \textsc{Eval}$$

Bottom-up proof search

apply **+**

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1 2

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n} + \texttt{n}\ \{\texttt{n} \mapsto 5; 10\}$

Bottom-up proof search

apply **+**

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1

$\{emp\}\ \texttt{1}\ \{emp; 1\}$ 2

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n} + \texttt{n}\ \{\texttt{n} \mapsto 5; 10\}$

Bottom-up proof search

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1

$\{emp\}\ \texttt{1}\ \{emp; 1\}$ 2

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n + n}\ \{\texttt{n} \mapsto 5; 10\}$

apply **+**

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$

Bottom-up proof search

apply +

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1

$\{emp\}\ \texttt{1}\ \{emp; 1\}$ 2

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ +\ \texttt{n}\ \{\texttt{n} \mapsto 5; 10\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\} \quad \dfrac{\{emp\}\ \texttt{1}\ \{emp; 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5; 1\}}\ \textsc{Frame}$

Bottom-up proof search

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ 1

$\{emp\}\ \texttt{1}\ \{emp; 1\}$ 2

$\{emp\}\ \texttt{0}\ \{emp; 0\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n + n}\ \{\texttt{n} \mapsto 5; 10\}$

apply **+**

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\} \quad \dfrac{\{emp\}\ \texttt{1}\ \{emp; 1\}}{\{\texttt{n} \mapsto 5\}\ \texttt{1}\ \{\texttt{n} \mapsto 5; 1\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n + 1}\ \{\texttt{n} \mapsto 5; 6\}}\ \textsc{Eval}$$

Bottom-up proof search

apply **+**

$\{\mathtt{arr} \mapsto v_{arr}\}\ \mathtt{arr}\ \{\mathtt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\mathtt{n} \mapsto 5\}\ \mathtt{n}\ \{\mathtt{n} \mapsto 5; 5\}$ (1)

$\{emp\}\ \mathtt{1}\ \{emp; 1\}$ (2)

$\{emp\}\ \mathtt{0}\ \{emp; 0\}$

$\{\mathtt{n} \mapsto 5\}\ \mathtt{n}\ +\ \mathtt{n}\ \{\mathtt{n} \mapsto 5; 10\}$

$\{\mathtt{n} \mapsto 5\}\ \mathtt{n}\ +\ \mathtt{1}\ \{\mathtt{n} \mapsto 5; 6\}$

Bottom-up proof search

apply **+**

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ (1)

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

$\{emp\}\ \texttt{0}\ \{emp; 0\}$ (2)

$\{\texttt{n} \mapsto 5\}\ \texttt{n + n}\ \{\texttt{n} \mapsto 5; 10\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n + 1}\ \{\texttt{n} \mapsto 5; 6\}$

Bottom-up proof search

apply **+**

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\, v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\}$ 1

$\{emp\}\ \texttt{1}\ \{emp;\, 1\}$

$\{emp\}\ \texttt{0}\ \{emp;\, 0\}$ 2

$\{\texttt{n} \mapsto 5\}\ \texttt{n + n}\ \{\texttt{n} \mapsto 5;\, 10\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n + 1}\ \{\texttt{n} \mapsto 5;\, 6\}$

$\{emp\}\ \texttt{0}\ \{emp;\, 0\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\}$

Bottom-up proof search

apply +

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr};\, v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\}$ (1)

$\{emp\}\ \texttt{1}\ \{emp;\, 1\}$

$\{emp\}\ \texttt{0}\ \{emp;\, 0\}$ (2)

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ +\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 10\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ +\ \texttt{1}\ \{\texttt{n} \mapsto 5;\, 6\}$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5;\, 5\} \qquad \dfrac{\{emp\}\ \texttt{0}\ \{emp;\, 0\}}{\{\texttt{n} \mapsto 5\}\ \texttt{0}\ \{\texttt{n} \mapsto 5;\, 0\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ +\ \texttt{0}\ \{\texttt{n} \mapsto 5;\, 5\}}\ \textsc{Eval}$$

Bottom-up proof search

$\{\mathtt{arr} \mapsto v_{arr}\}\ \mathtt{arr}\ \{\mathtt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\mathtt{n} \mapsto 5\}\ \mathtt{n}\ \{\mathtt{n} \mapsto 5; 5\}$ 1

$\{emp\}\ \mathtt{1}\ \{emp; 1\}$

$\{emp\}\ \mathtt{0}\ \{emp; 0\}$ 2

$\{\mathtt{n} \mapsto 5\}\ \mathtt{n} + \mathtt{n}\ \{\mathtt{n} \mapsto 5; 10\}$

$\{\mathtt{n} \mapsto 5\}\ \mathtt{n} + \mathtt{1}\ \{\mathtt{n} \mapsto 5; 6\}$

apply **+**

$$\dfrac{\{\mathtt{n} \mapsto 5\}\ \mathtt{n}\ \{\mathtt{n} \mapsto 5; 5\} \qquad \dfrac{\{emp\}\ \mathtt{0}\ \{emp; 0\}}{\{\mathtt{n} \mapsto 5\}\ \mathtt{0}\ \{\mathtt{n} \mapsto 5; 0\}}\ \textsc{Frame}}{\{\mathtt{n} \mapsto 5\}\ \mathtt{n} + \mathtt{0}\ \{\mathtt{n} \mapsto 5; 5\}}\ \textsc{Eval}$$

Bottom-up proof search

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ (1)

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

$\{emp\}\ \texttt{0}\ \{emp; 0\}$ (2)

$\{\texttt{n} \mapsto 5\}\ \texttt{n} + \texttt{n}\ \{\texttt{n} \mapsto 5; 10\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n} + \texttt{1}\ \{\texttt{n} \mapsto 5; 6\}$

apply **+**

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\} \qquad \dfrac{\{emp\}\ \texttt{0}\ \{emp; 0\}}{\{\texttt{n} \mapsto 5\}\ \texttt{0}\ \{\texttt{n} \mapsto 5; 0\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n} + \texttt{0}\ \{\texttt{n} \mapsto 5; 5\}}\ \textsc{Eval}$$

Bottom-up proof search

apply **+**

$\{\texttt{arr} \mapsto v_{arr}\}\ \texttt{arr}\ \{\texttt{arr} \mapsto v_{arr}; v_{arr}\}$

$\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\}$ **1**

$\{emp\}\ \texttt{1}\ \{emp; 1\}$

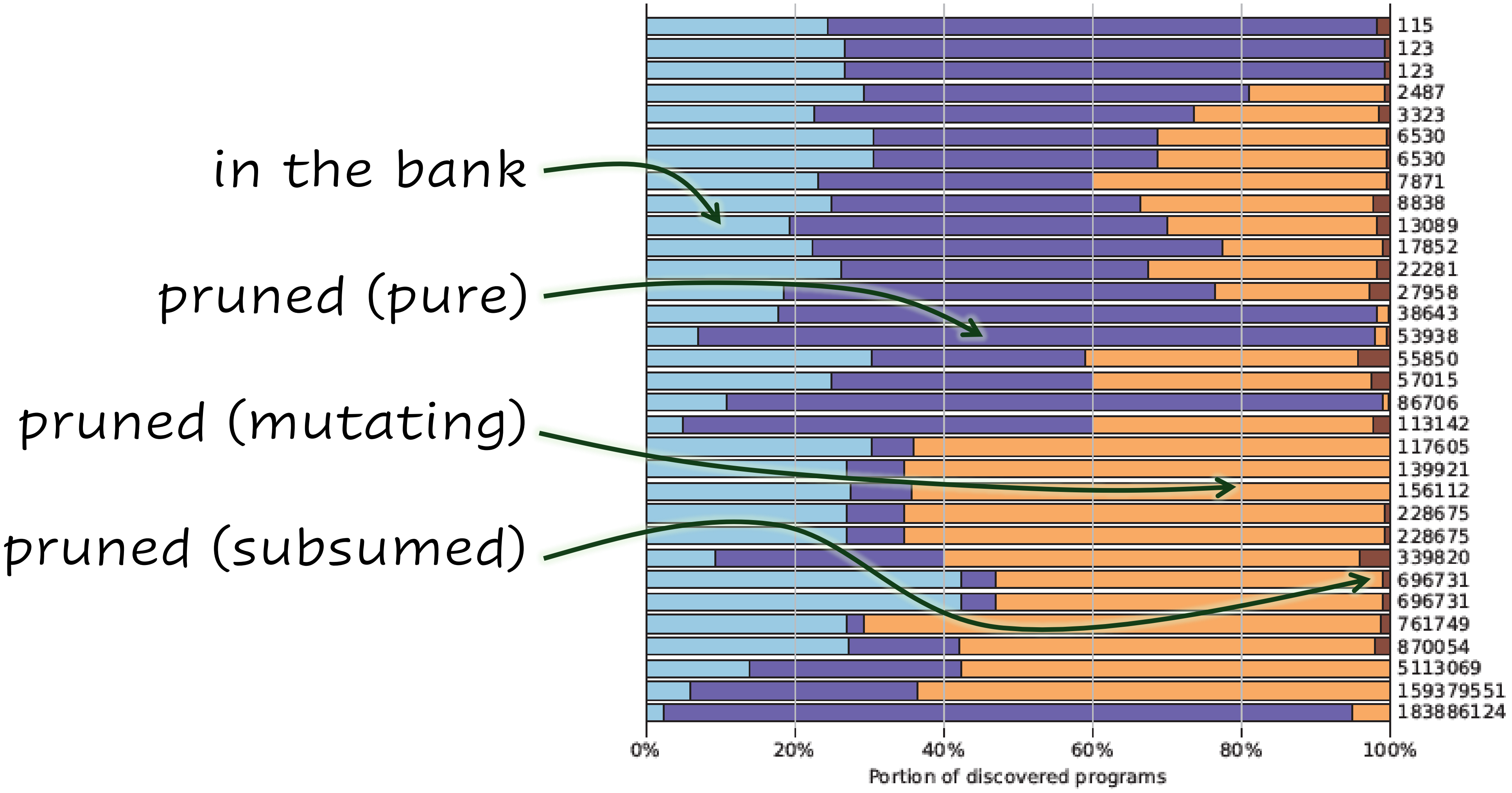
$\{emp\}\ \texttt{0}\ \{emp; 0\}$ **2**

$\{\texttt{n} \mapsto 5\}\ \texttt{n + n}\ \{\texttt{n} \mapsto 5; 10\}$

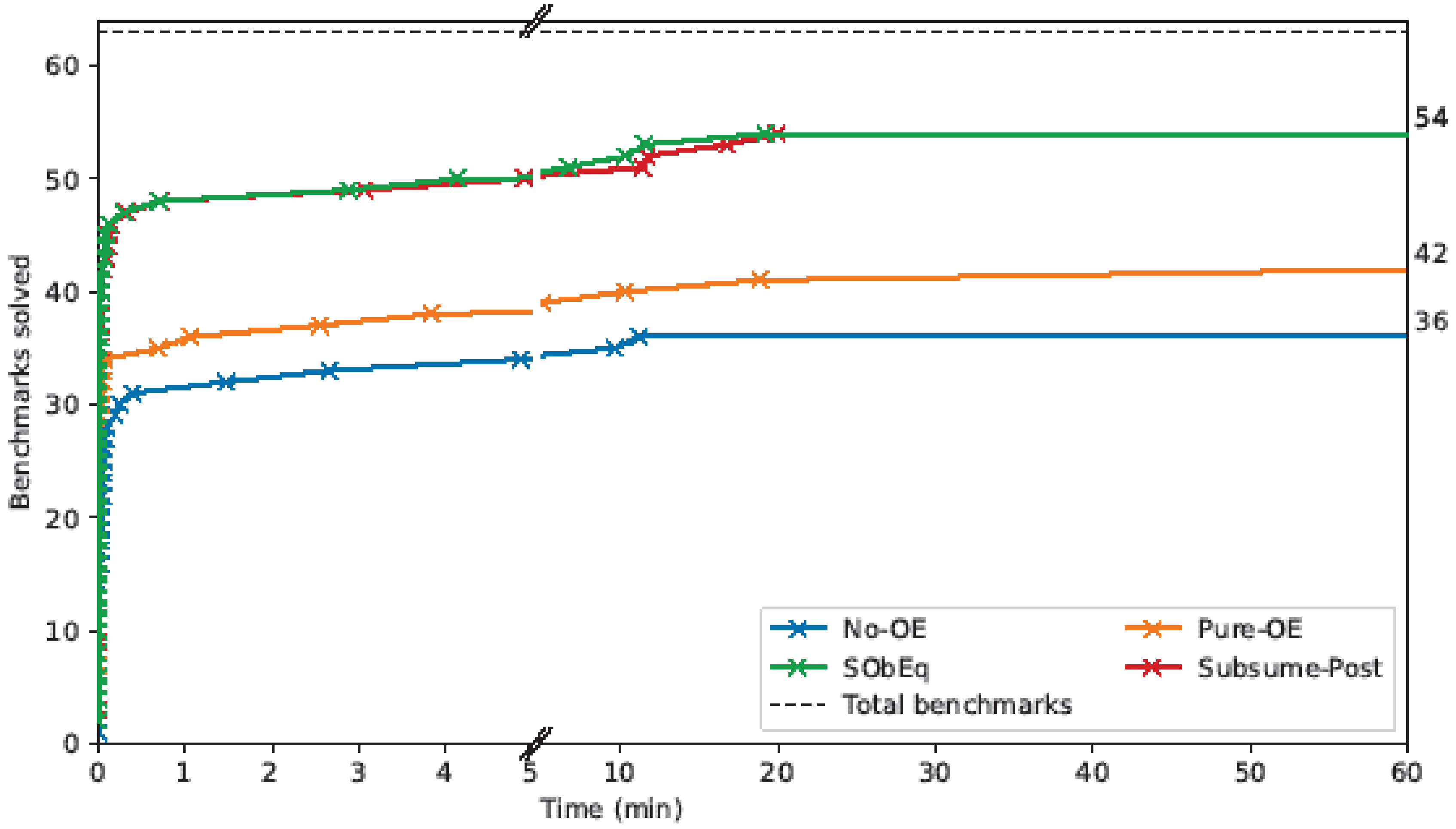
$\{\texttt{n} \mapsto 5\}\ \texttt{n + 1}\ \{\texttt{n} \mapsto 5; 6\}$

$$\dfrac{\{\texttt{n} \mapsto 5\}\ \texttt{n}\ \{\texttt{n} \mapsto 5; 5\} \qquad \dfrac{\{emp\}\ \texttt{0}\ \{emp; 0\}}{\{\texttt{n} \mapsto 5\}\ \texttt{0}\ \{\texttt{n} \mapsto 5; 0\}}\ \textsc{Frame}}{\{\texttt{n} \mapsto 5\}\ \texttt{n + 0}\ \{\texttt{n} \mapsto 5; 5\}}\ \textsc{Eval}$$

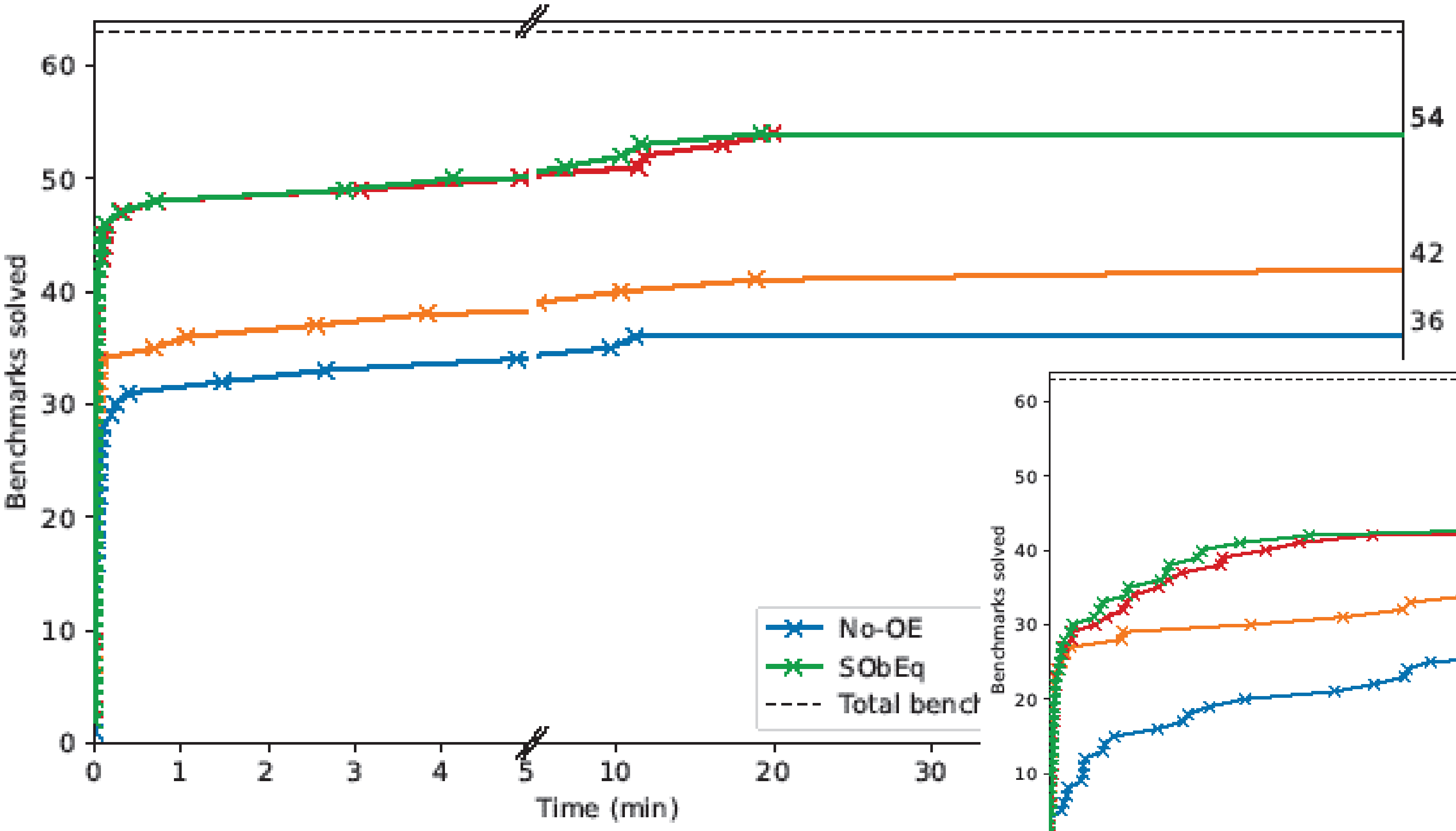
Results: pruned programs

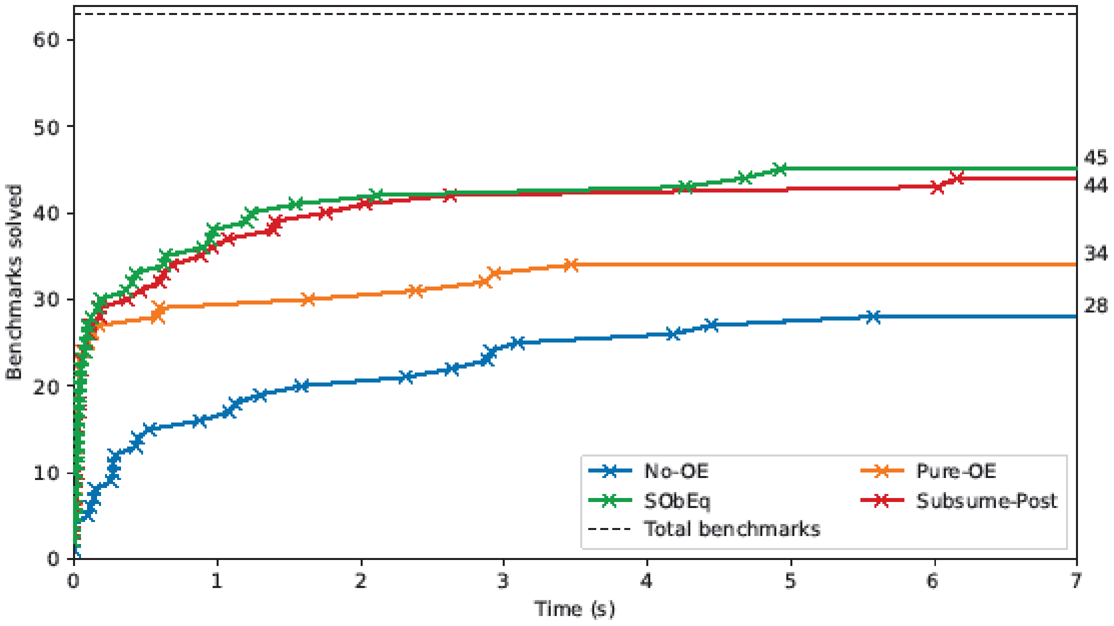


Results: Pruning gives us speed



Results: Pruning gives us speed





Results: determinism and quality

FrAngel: Shi et al. 2019

```
arr.splice(beg++,1,toadd).reverse();
return end + 1;
```

Results: determinism and quality

FrAngel: Shi et al. 2019

```
arr.splice(beg++,1,toadd).reverse();
return end + 1;
```

return temp array

Results: determinism and quality

FrAngel: Shi et al. 2019

```
arr.splice(beg++,1,toadd).reverse();
return end + 1;
```

return temp array

reverse & discard

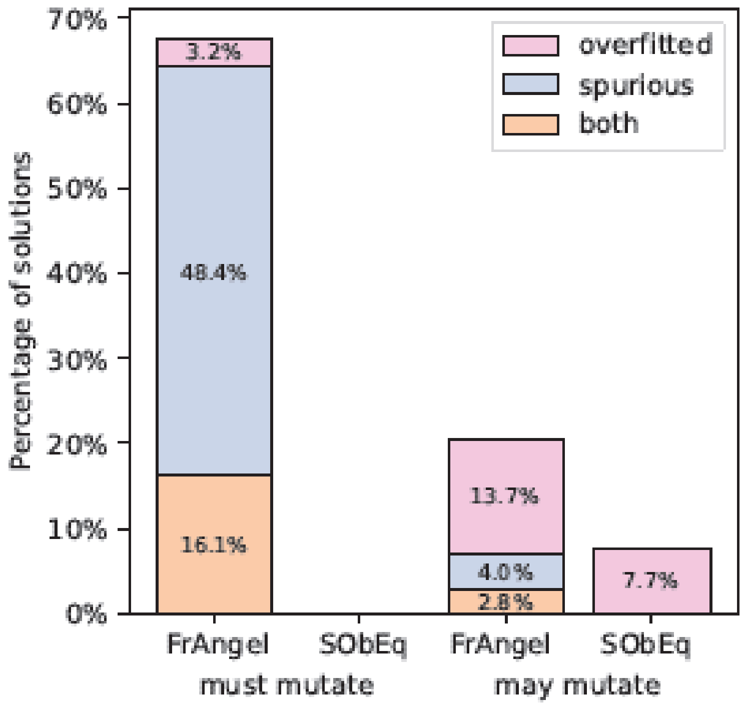
Results: determinism and quality

FrAngel: Shi et al. 2019

```
arr.splice(beg++,1,toadd).reverse();
return end + 1;
```

return temp array

reverse & discard



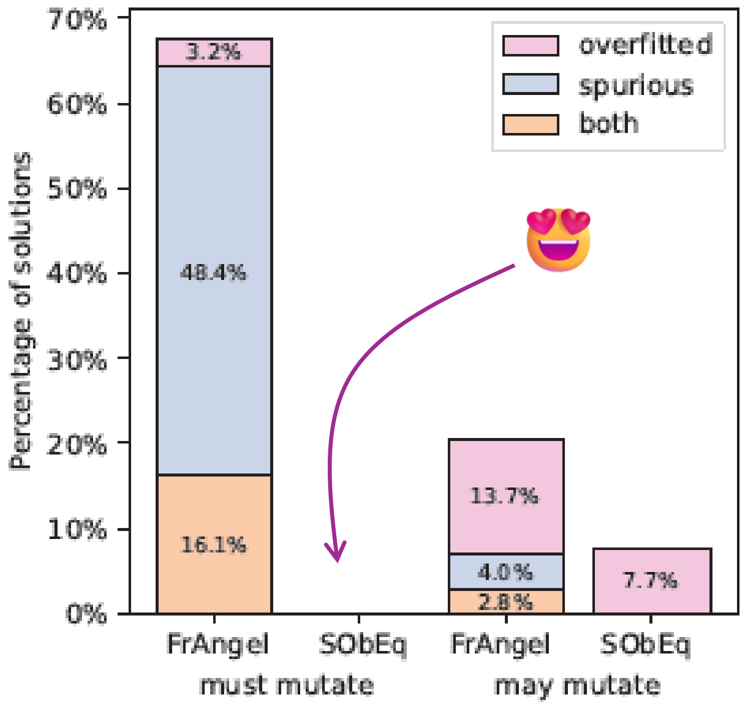
Results: determinism and quality

FRANGEL: Shi et al. 2019

```
arr.splice(beg++,1,toadd).reverse();
return end + 1;
```

return temp array

reverse & discard



***S**ide-effects in **OB**servational **EQ**uivalence*

1) Change representation to triples

$$\underbrace{\{x \mapsto v_x * \ y \mapsto v_y, \dots\}}_{\text{precondition}}\ p\ \underbrace{\{x \mapsto v_x' * y \mapsto v_y', \dots ; r\}}_{\text{postcondition}}$$

precondition

postcondition

result

2) Combine them using Separation Logic

$$\frac{\{P_1\}\, p_1\, \{P_2; r_1\} \quad \{P_2\}\, p_2\, \{P_3; r_2\} \quad \cdots \quad \{P_k\}\, p_k\, \{P_{k+1}; r_k\} \qquad (c(r_1, \dots, r_k), P_{k+1}) \to (r, Q)}{\{P_1\}\, c(p_1, \dots, p_k)\, \{Q; r\}}\ \textsc{Eval}$$

$$\frac{\{P\}\, p\, \{Q; r\}}{\{P * R\}\, p\, \{Q * R; r\}}\ \textsc{Frame}$$

Result: correct enumeration with mutations!