Static Analysis for Data Science

MPRI 2-6: Abstract Interpretation, Application to Verification and Static Analysis

Caterina Urban

January 20th, 2024

Year 2024-2025

Machine Learning Development Pipeline

TODAY

PREVIOUS COURSE

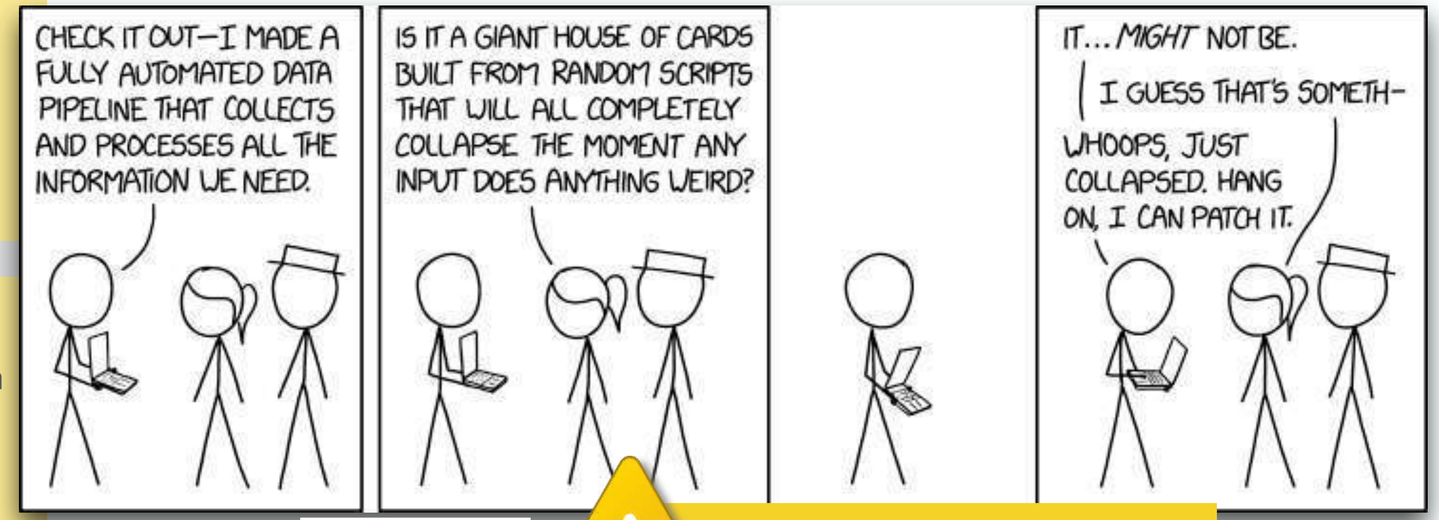
Static Analysis for Data Preparation

Machine Learning Development Pipeline

Data Preparation is Fragile

data

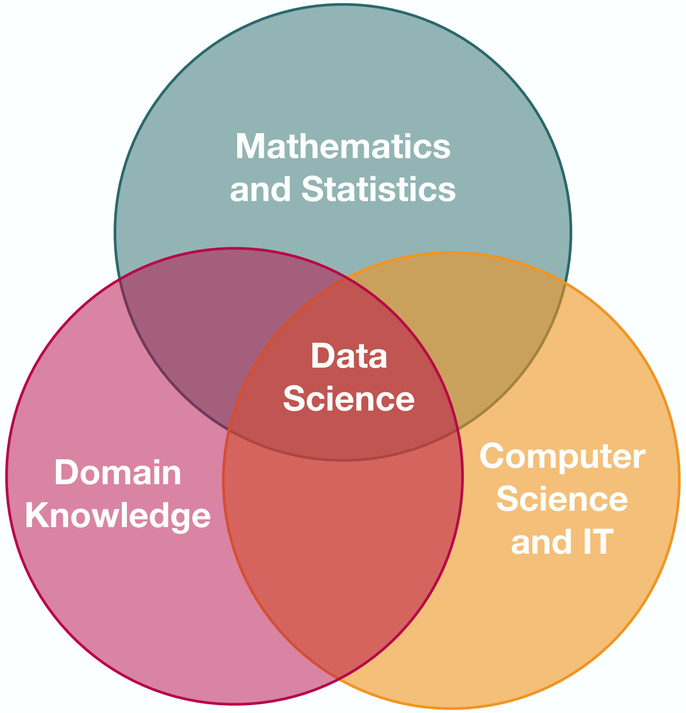
data preparation



https://xkcd.com/2054/

insidious silent bugs

Data Scientists



https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

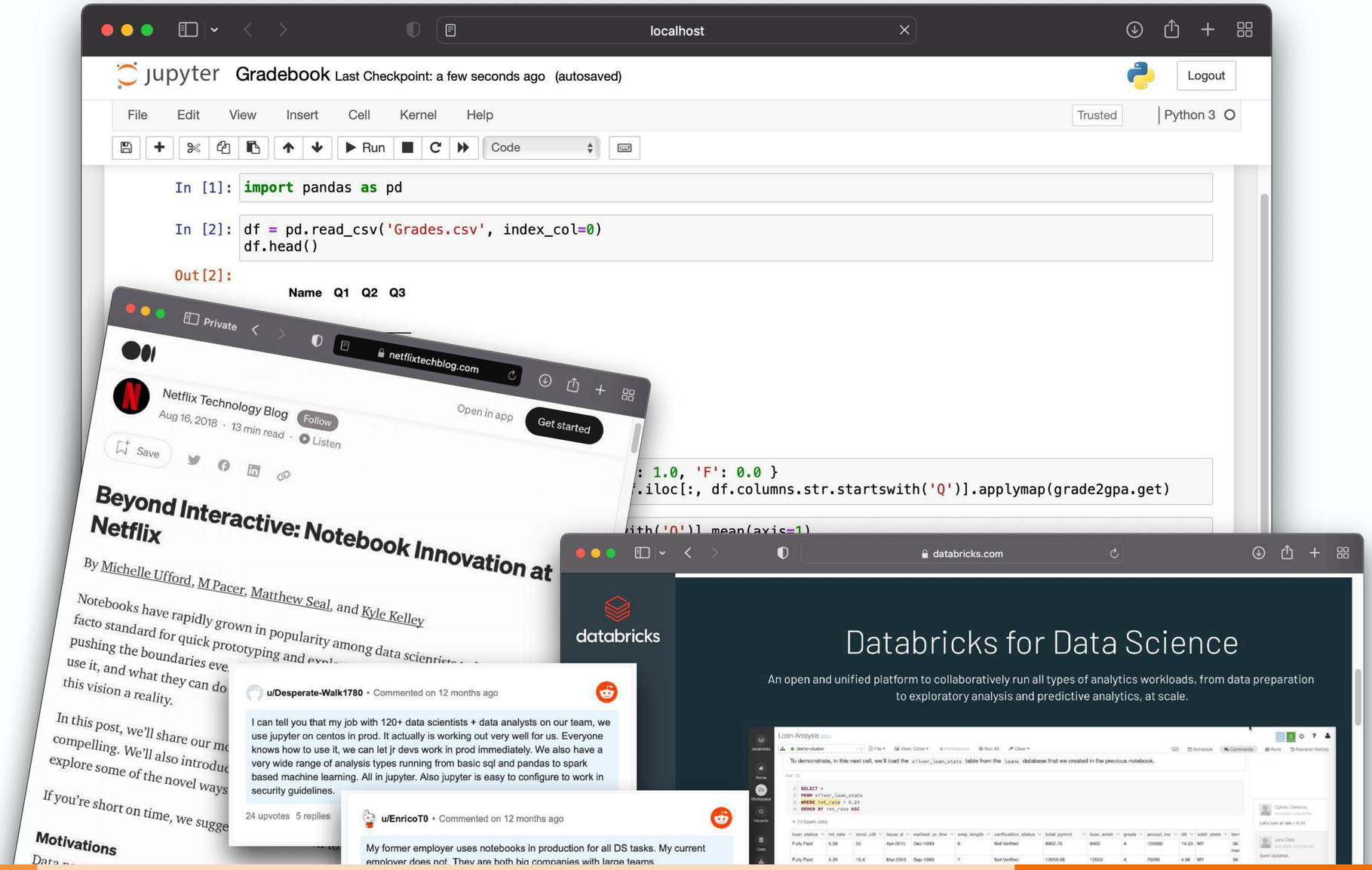
Data Scientist: The Sexiest Job of the 21st Century

Andrew McAfee and Erik Brynjolfsson

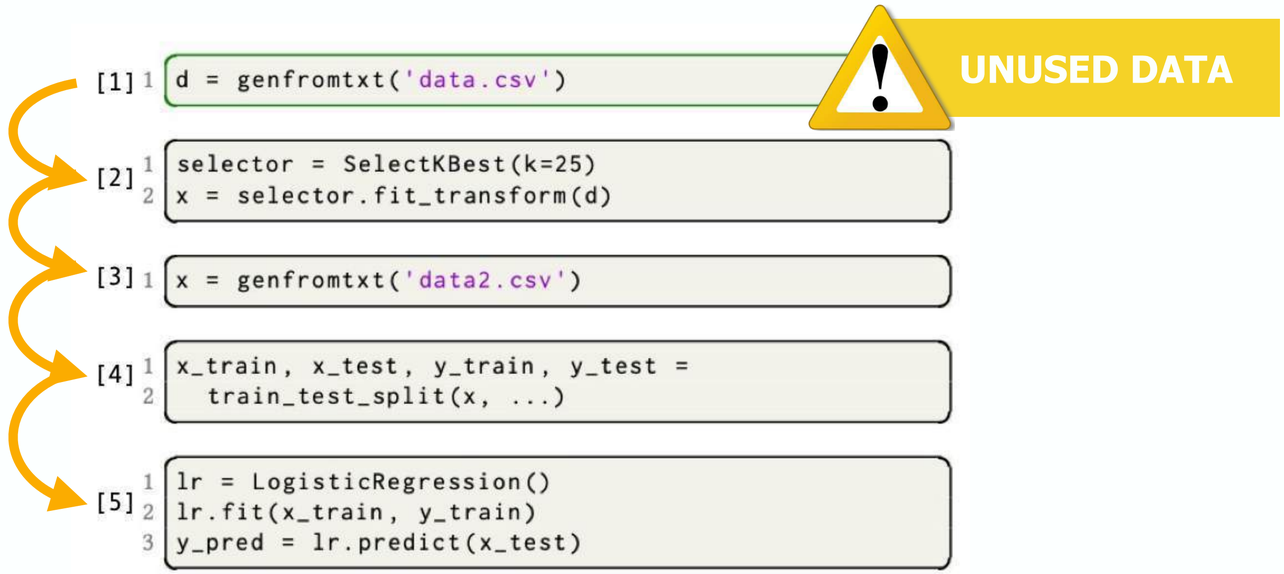
Andrew J Buboltz, silk screen on a page from a high school yearbook, 8.5" x 12", 2011 Tamar Cohen

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know

Jupyter Notebooks

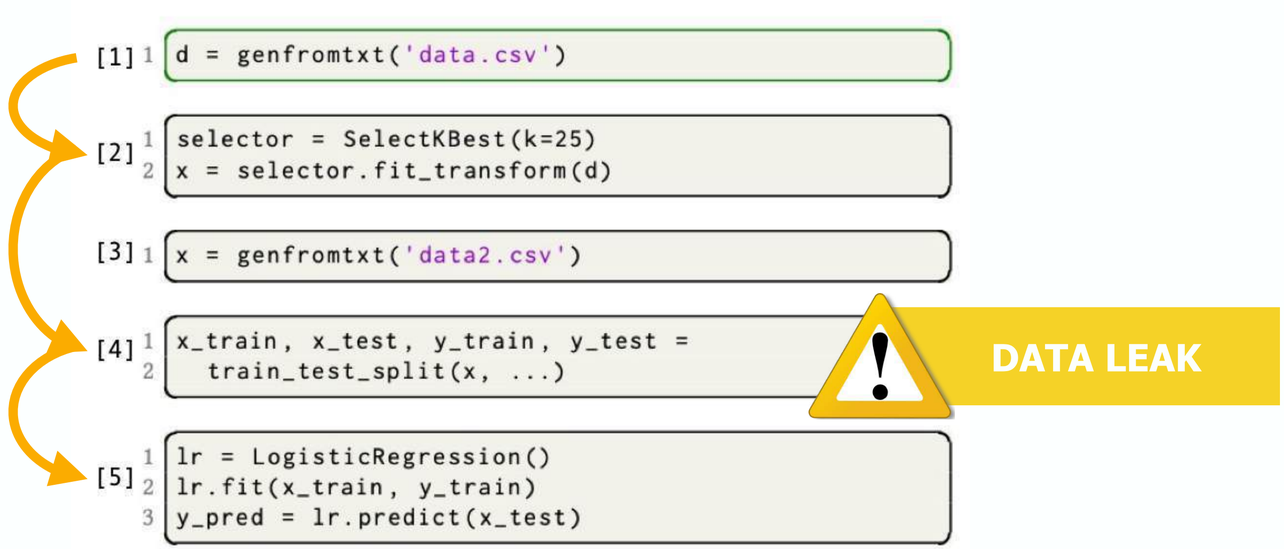


Jupyter Notebooks



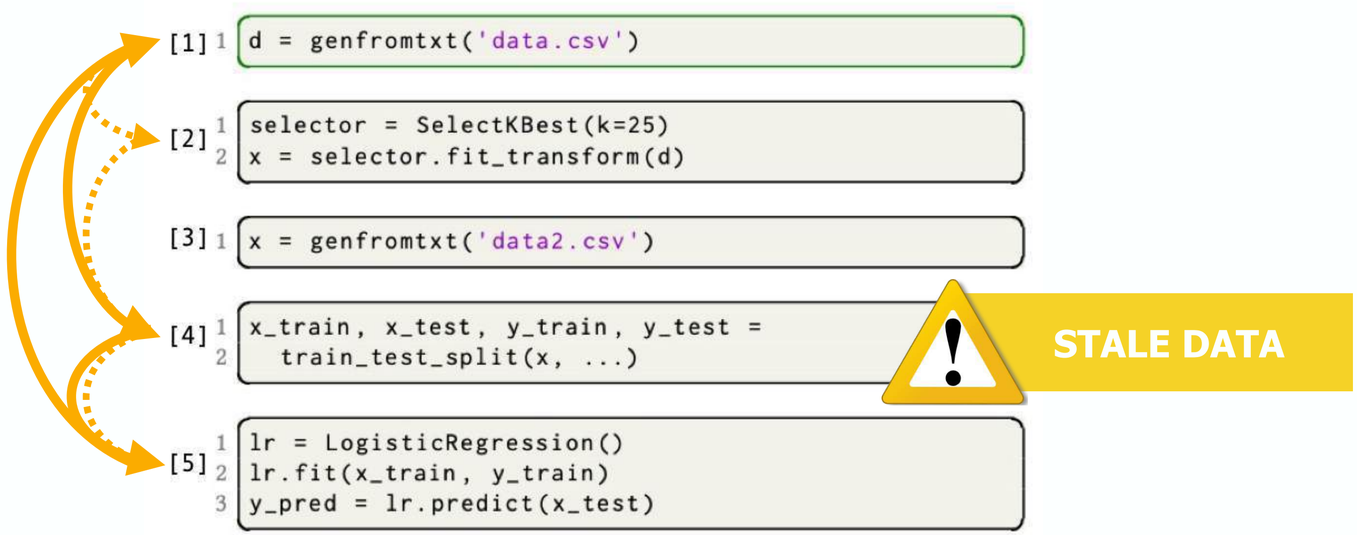
P. Subotić et al. - A Static Analysis Framework for Data Science Notebooks (ICSE 2022)

Jupyter Notebooks



P. Subotić et al. - A Static Analysis Framework for Data Science Notebooks (ICSE 2022)

Jupyter Notebooks



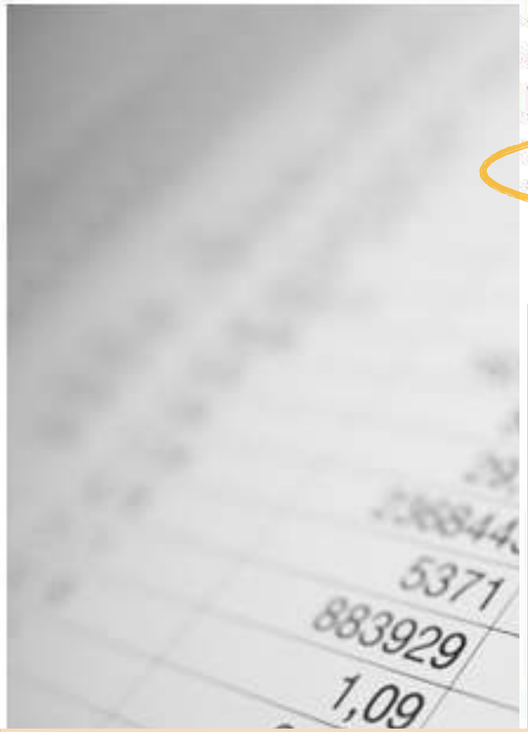
P. Subotić et al. - A Static Analysis Framework for Data Science Notebooks (ICSE 2022)

Anomalously Unused Data

The Reinhart-Rogoff Paper

FAQ: Reinhart, Rogoff, and the Excel Error That Changed History

By Peter Coy | April 18, 2013



The Excel Depression

By PAUL KRUGMAN
Published: April 18, 2013 | 470 Comments

In this age of information, math errors can lead to disaster. NASA's Mars Orbiter crashed because engineers forgot to convert to metric measurements; JPMorgan Chase's "London Whale" venture went bad in part because modelers divided by a sum instead of an average. So, did an Excel coding error destroy the economies of the Western world?

Enlarge This Image

The story so far: At the beginning of 2010, two Harvard economists, Carmen Reinhart and Kenneth Rogoff, circulated a paper, "Growth in a Time of Debt," that purported to identify a critical "threshold," a tipping point, for government indebtedness. Once debt exceeds 90 percent of gross domestic product, they claimed, economic growth drops off sharply.

Ms. Reinhart and Mr. Rogoff had credibility thanks to a

FACEBOOK
TWITTER
GOOGLE+
SAVE
EMAIL
SHARE
PRINT
REPRINTS

England Covid-19 Cases Error

SCIENCE | US & WORLD | TECH

Excel spreadsheet error blamed for UK's 16,000 missing coronavirus cases

The case went missing after the spreadsheet hit its filesize limit

By James Vincent | Oct 5, 2020, 9:41am EDT

NEWS

Check for updates

The BMJ

Cite this as: *BMJ* 2020;371:m3891

http://dx.doi.org/10.1136/bmj.m3891

Published: 06 October 2020

Covid-19: Only half of 16 000 patients missed from England's official figures have been contacted

Elisabeth Mahase

Details of nearly 16 000 cases of covid-19 were not transferred to England's NHS Test and Trace service and were missed from official figures because of an error in the process for updating the data.

England's health and social care secretary, Matt Hancock, told the House of Commons on Monday 5 October that after the error was discovered on Friday 2 October "6500 hours of extra contact tracing" had been carried out over the weekend. But as at Monday morning only half (51%) of the people had been reached by contact tracers.

...nse. Labour's shadow health secretary, ...id. "Thousands of people are ...sed to covid,

data and furthermore have issued guidance on validation and risk management for these products if they are to be used in such a safety critical manner."

The error came as the Labour Party's leader, Keir Starmer, said that the prime minister had "lost control" of covid-19, with no clear strategy for beating it. Speaking to the *Observer*, Starmer set out his five point plan for covid-19, which starts with publishing the criteria for local restrictions, as the German government did. Secondly, he said public health messaging should be improved by adding a feature to the NHS covid-19 app so people can search their postcode and find out their local restrictions.

Starmer has also said he would fix the contact tracing ...ystem by investing in NHS and university ...expand testing and at the same time ...in charge of contact ...high

BMJ: first published as 10.1136/bmj.m3891 on 6 October 2020. D

Example

```
english = bool(input())
math = bool(input())
science = bool(input())
bonus = bool(input())

passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus

print(passing)
```

INPUT VARIABLES

ERROR: english SHOULD BE passing

ERROR: math SHOULD BE science

OUTPUT VARIABLES

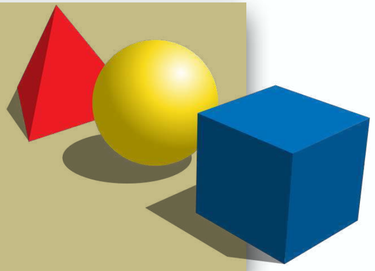
the input variables english and science are unused

Data Usage Static Analysis [Urban18]

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Dependency Fairness

$$\mathscr{F}_i \stackrel{\text{def}}{=} \{[\![M]\!] \mid \text{UNUSED}_i([\![M]\!])\}$$

$\mathscr{F}_i$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **do not use** the value of the sensitive input node $x_{0,i}$ for classification

$$\text{UNUSED}_i(T) \stackrel{\text{def}}{=} \forall t, t' \in T: t_0(x_{0,i}) \neq t'_0(x_{0,i}) \wedge \eta(t_0) = \eta(t'_0)$$
$$(\forall 0 \leq j \leq |L_0| : j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j}))$$
$$\Rightarrow t_\omega = t'_\omega \quad \rho(t_0) = \rho(t'_0)$$

η:

$$\eta(x0j) = \begin{cases} \top & j = i \\ x0j & \text{otherwise} \end{cases}$$

Intuitively: inputs differing only on the value of the sensitive input node $x_{0,i}$ should lead to the same **classification outcome**

Theorem

$$M \vDash \mathscr{F}_i \Leftrightarrow \{[\![M]\!]\} \subseteq \mathscr{F}_i$$

Corollary

$$M \vDash \mathscr{F}_i \Leftarrow [\![M]\!] \subseteq [\![M]\!]^\natural \subseteq \mathscr{F}_i$$

Data (Non-)Usage

$$\mathcal{N}_J \overset{\text{def}}{=} \{ [\![P]\!] \mid \text{UNUSED}_J([\![P]\!]) \}$$

$\mathcal{N}_J$ is the set of all programs P (or, rather, their semantics $[\![P]\!]$) that **do not use** the value of the input variables in J

$$\begin{aligned}\text{UNUSED}_J([\![P]\!]) \overset{\text{def}}{=} \; & \forall t \in [\![P]\!], V \in \mathscr{R}^{|J|} : t_0(J) \neq V \Rightarrow \exists t' \in [\![P]\!] : \\ & (\forall i : i \notin J \Rightarrow t_0(i) = t'_0(i)) \\ & \wedge\, t'_0(J) = V \\ & \wedge\, t_\omega = t'_\omega\end{aligned}$$

Intuitively: **any possible program outcome** is possible **from any value of the input variable** i

Theorem

$$P \vDash \mathcal{N}_J \Leftrightarrow \{ [\![P]\!] \} \subseteq \mathcal{N}_J$$

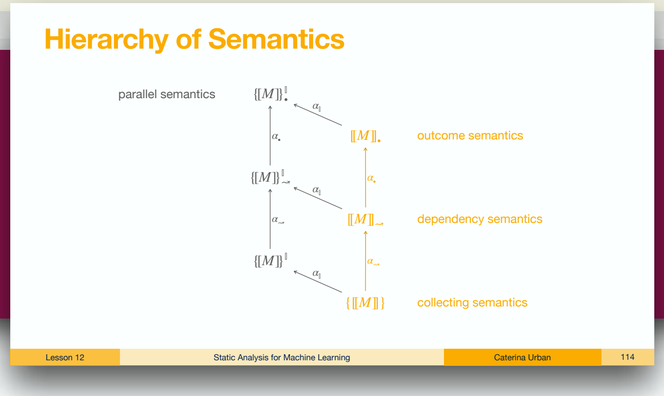
Data Usage Static Analysis [Urban18]

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

concrete semantics
mathematical models of the program behavior

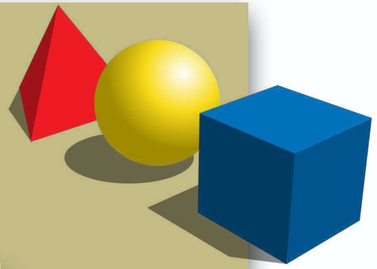


Data Usage Static Analysis [Urban18]

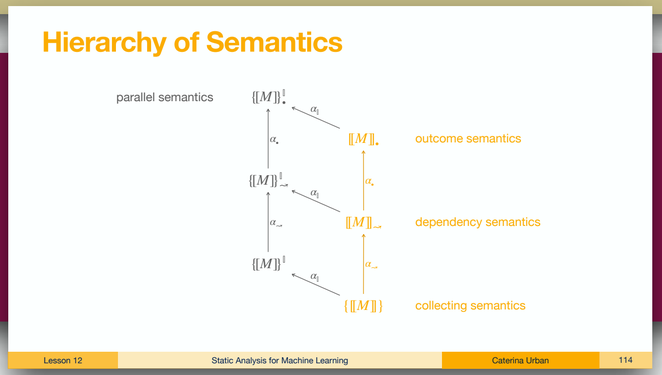
3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

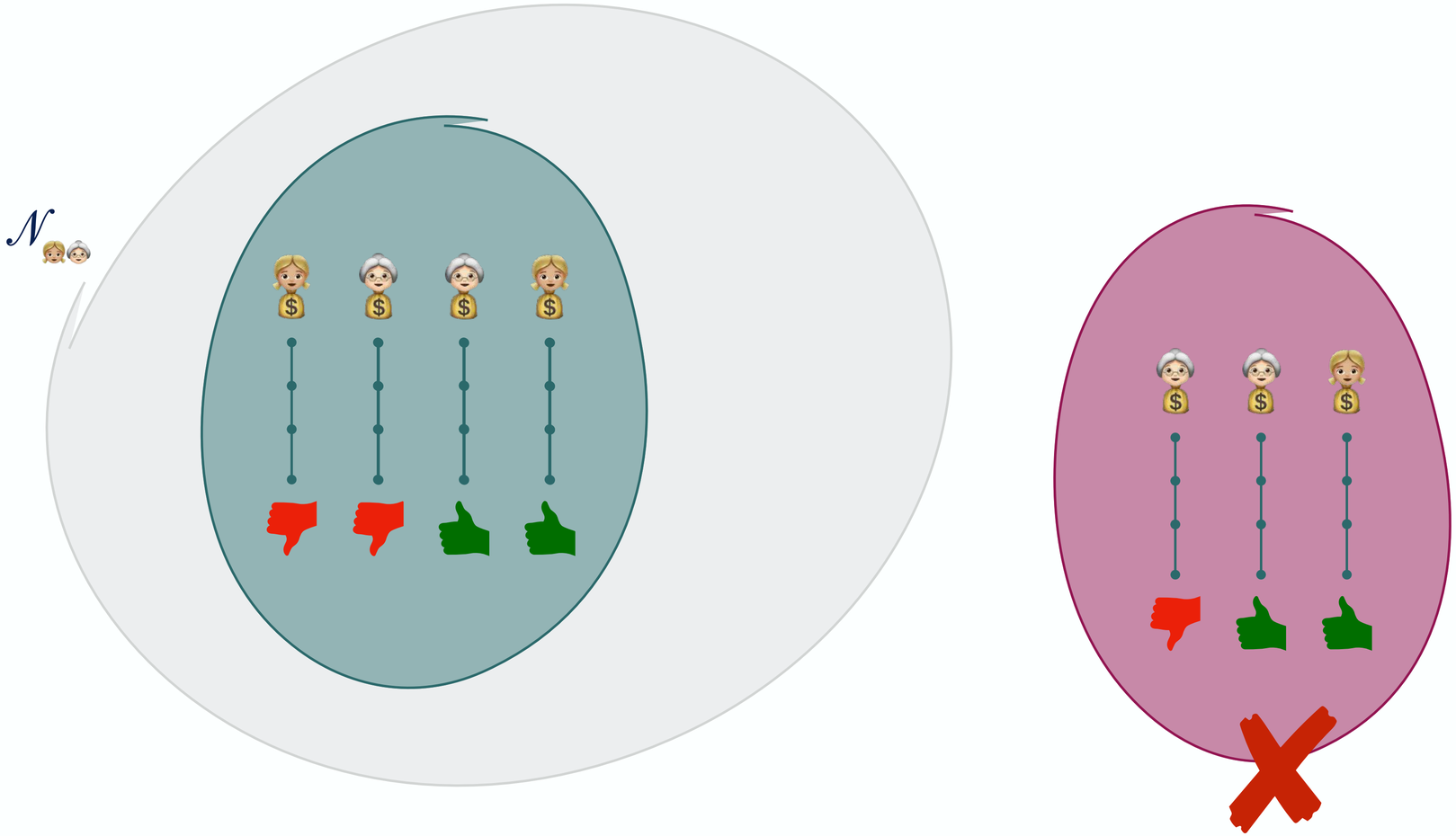


concrete semantics
mathematical models of the program behavior



Data (Non-) Usage

Not a Subset-Closed Property



Data (Non-)Usage

$$\mathcal{N}_J \stackrel{\text{def}}{=} \{ [\![P]\!] \mid \text{UNUSED}_J([\![P]\!]) \}$$

$\mathcal{N}_J$ is the set of all programs P (or, rather, their semantics $[\![P]\!]$) that **do not use** the value of the input variables in J

$$\begin{aligned}\text{UNUSED}_J([\![P]\!]) \stackrel{\text{def}}{=} \; & \forall t \in [\![P]\!], V \in \mathscr{R}^{|J|} : t_0(J) \neq V \Rightarrow \exists t' \in [\![P]\!] : \\ & (\forall i : i \notin J \Rightarrow t_0(i) = t'_0(i)) \\ & \wedge\, t'_0(J) = V \\ & \wedge\, t_\omega = t'_\omega\end{aligned}$$

Intuitively: **any possible program outcome** is possible **from any value of the input variable** i

Theorem

$$P \vDash \mathcal{N}_J \Leftrightarrow \{ [\![P]\!] \} \subseteq \mathcal{N}_J$$

Corollary

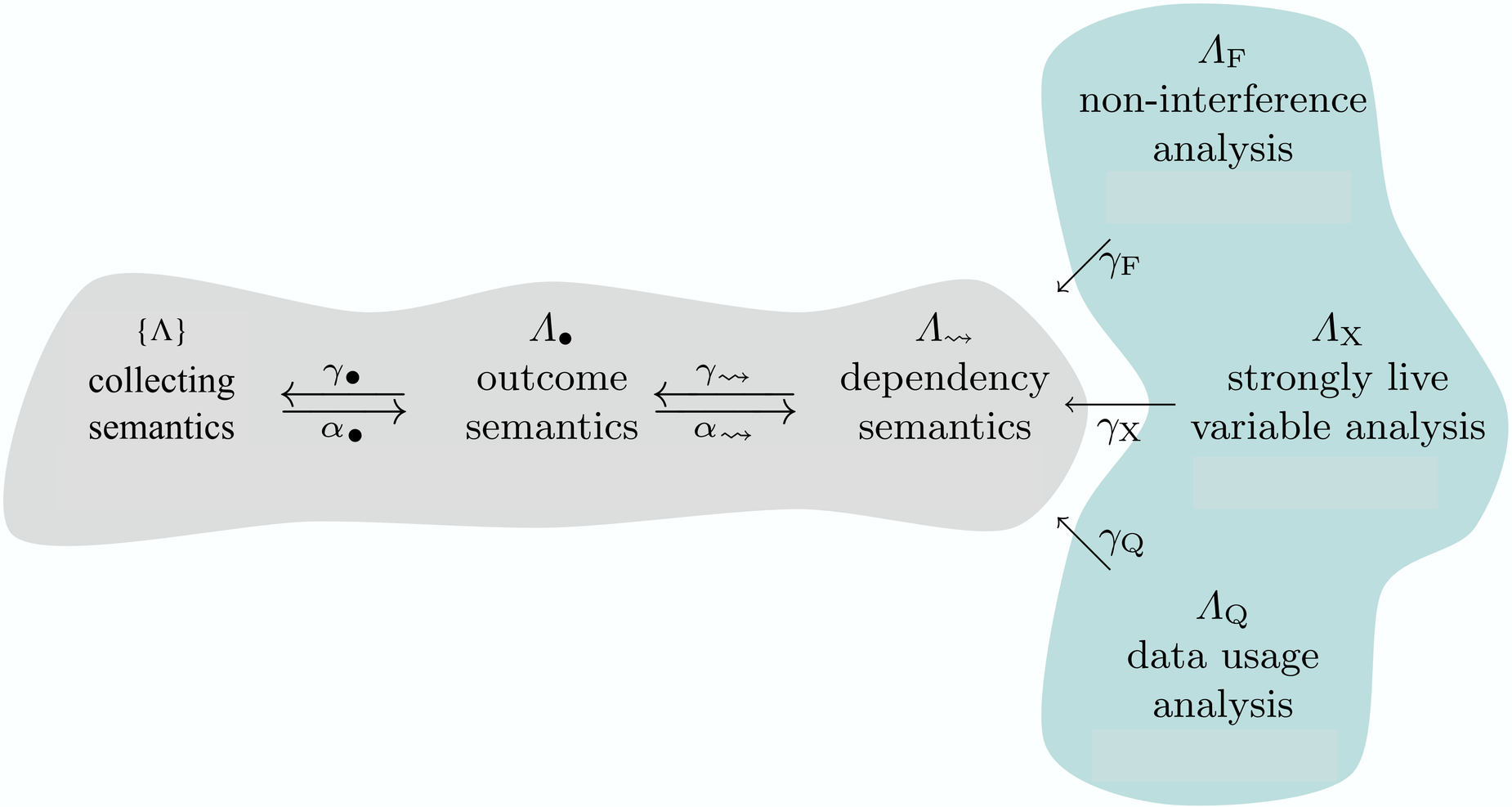
$$P \vDash \mathcal{N}_J \Leftarrow \{ [\![P]\!] \} \subseteq [\![P]\!]^\natural \in \mathcal{N}_J$$

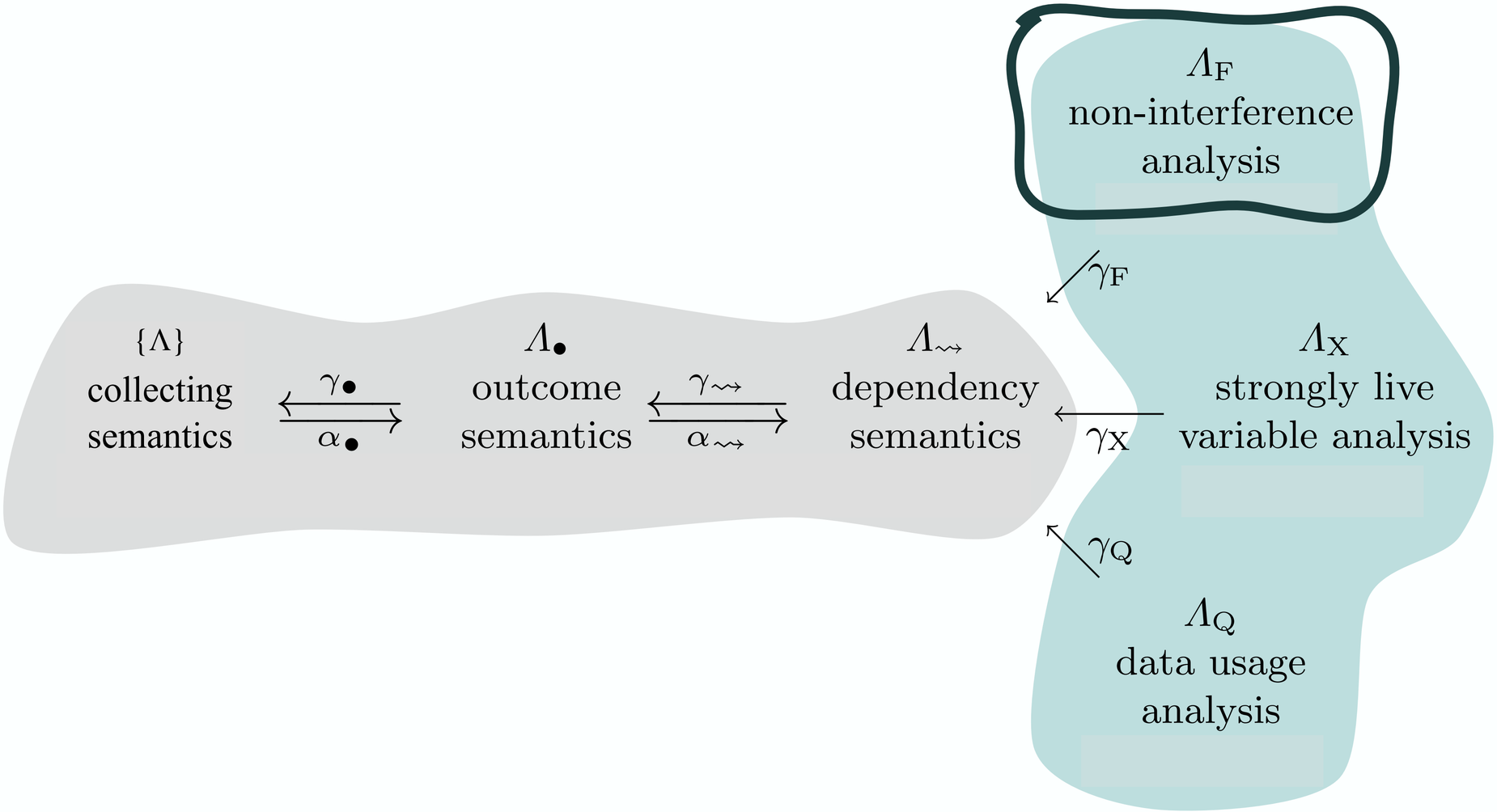
Data (Non-)Usage Abstractions

Over-Approximation of the Used Input Data

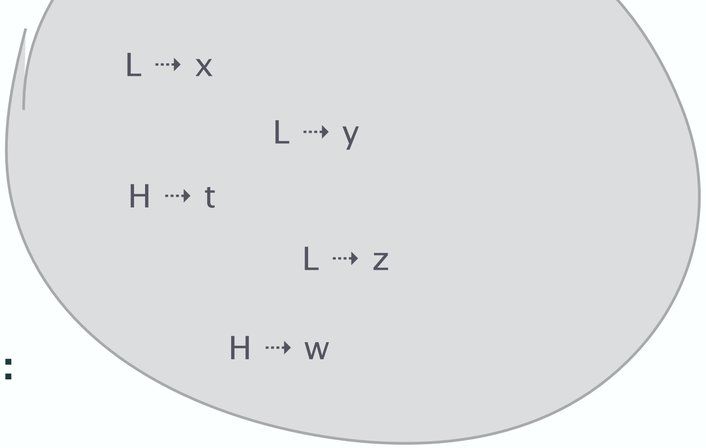
$\Rightarrow$ Under-Approximation of the Unused Input Data

$$P \vDash \mathcal{N}_{J^\natural \subseteq J} \Leftarrow \{[\![P]\!]\} \subseteq [\![P]\!]^\natural_A \subseteq \mathcal{N}_{J^\natural \subseteq J}$$


$\{\Lambda\}$
collecting semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome semantics
$\gamma_{\rightsquigarrow}$
$\alpha_{\rightsquigarrow}$
$\Lambda_{\rightsquigarrow}$
dependency semantics
γ_X
Λ_F
non-interference analysis
γ_F
Λ_X
strongly live variable analysis
γ_Q
Λ_Q
data usage analysis


Λ_{F}
non-interference
analysis
γ_{F}
$\{\Lambda\}$
collecting
semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome
semantics
$\gamma_{\rightsquigarrow}$
$\alpha_{\rightsquigarrow}$
$\Lambda_{\rightsquigarrow}$
dependency
semantics
γ_{X}
Λ_{X}
strongly live
variable analysis
γ_{Q}
Λ_{Q}
data usage
analysis

Secure Information Flow



possibilistic non-interference coincides with input data (non-)usage when the set J of unused input variables contains *all* input variables:

- **input variables are high-security variables**
- **output variables are low-security variables**

$[\![P]\!]_{\mathrm{F}}$

explicit usage flows

implicit usage flows

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\Theta_{\mathrm{F}}[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$

$$\Theta_{\mathrm{F}}[\![x = e]\!](S) \stackrel{\text{def}}{=} \{L \rightsquigarrow y \in S \mid y \neq x\} \cup \{L \rightsquigarrow x \mid \mathcal{V}_{\mathrm{F}}[\![e]\!]S\}$$

$$\Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mathrm{F}}[\![s_1]\!](S) \sqcup_{\mathrm{F}} \Theta_{\mathrm{F}}[\![s_2]\!](S) & \text{if } \mathcal{V}_{\mathrm{F}}[\![e]\!]S \\ \{L \rightsquigarrow x \in S \mid x \notin \mathrm{w}(s_1) \cup \mathrm{w}(s_2)\} & \text{otherwise} \end{cases}$$

$$\Theta_{\mathrm{F}}[\![\texttt{while}\ e\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \mathrm{lfp}^{\sqsubseteq_{\mathrm{F}}}_{S}\ \Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]$$

$$\Theta_{\mathrm{F}}[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_{\mathrm{F}}[\![s_2]\!] \circ \Theta_{\mathrm{F}}[\![s_1]\!](S)$$

S guarantees a unique value for e independently of values of input variables

$\mathcal{V}_{\mathrm{F}}[\![x]\!]S \Leftrightarrow L \rightsquigarrow x \in S$

set of variables modified by s_i

$\mathcal{L} \stackrel{\text{def}}{=} \{L, H\}$: set of security levels

$L \rightsquigarrow x$: dependency constraint

$\mathrm{F} \stackrel{\text{def}}{=} \{L \rightsquigarrow x \mid x \in X\}$

$\langle \wp(\mathrm{F}), \sqsubseteq_{\mathrm{F}}, \sqcup_{\mathrm{F}} \rangle$: abstract domain

$S_1 \sqsubseteq_{\mathrm{F}} S_2 \stackrel{\text{def}}{=} S_1 \supseteq S_2$

$S_1 \sqcup_{\mathrm{F}} S_2 \stackrel{\text{def}}{=} S_1 \cap S_2$

Hypercollecting Semantics and Its Application to Static Analysis of Information Flow

Mounir Assaf
Stevens Institute of Technology, Hoboken, US
first.last@stevens.edu

David A. Naumann
Stevens Institute of Technology, Hoboken, US
first.last@stevens.edu

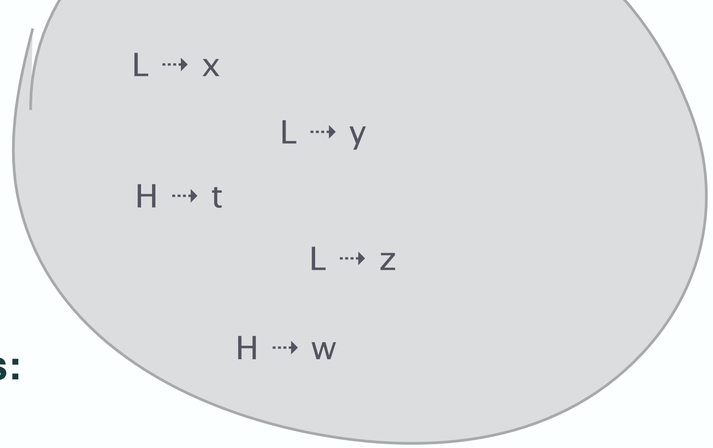
Julien Signoles
Software Reliability and Security Lab, CEA LIST, Saclay, FR
first.last@cea.fr

Éric Totel
CIDRE, CentraleSupélec, Rennes, FR
first.last@centralesupelec.fr

Frédéric Tronel
CIDRE, CentraleSupélec, Rennes, FR
first.last@centralesupelec.fr

program is correct if all its traces satisfy the predicate. By contrast with such *trace properties*, extensional definitions of dependences ore than one trace. To express that the final value of a ly on the initial value of a variable y, the in the security literature

Secure Information Flow



$[\![P]\!]_{\mathrm{F}}$

possibilistic non-interference coincides with input data (non-)usage when the set J of unused input variables contains *all* input variables:

- **input variables are high-security variables**
- **output variables are low-security variables**

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\Theta_{\mathrm{F}}[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$

$$\Theta_{\mathrm{F}}[\![x = e]\!](S) \stackrel{\text{def}}{=} \{L \rightsquigarrow y \in S \mid y \neq x\} \cup \{L \rightsquigarrow x \mid \mathcal{V}_{\mathrm{F}}[\![e]\!]S\}$$

$$\Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mathrm{F}}[\![s_1]\!](S) \sqcup_{\mathrm{F}} \Theta_{\mathrm{F}}[\![s_2]\!](S) & \text{if } \mathcal{V}_{\mathrm{F}}[\![e]\!]S \\ \{L \rightsquigarrow x \in S \mid x \notin \mathrm{w}(s_1) \cup \mathrm{w}(s_2)\} & \text{otherwise} \end{cases}$$

$$\Theta_{\mathrm{F}}[\![\texttt{while}\ e\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \mathrm{lfp}_S^{\sqsubseteq_{\mathrm{F}}}\ \Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]$$

$$\Theta_{\mathrm{F}}[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_{\mathrm{F}}[\![s_2]\!] \circ \Theta_{\mathrm{F}}[\![s_1]\!](S)$$

```
passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

L ⇢ passing, H ⇢ english, math, science, bonus

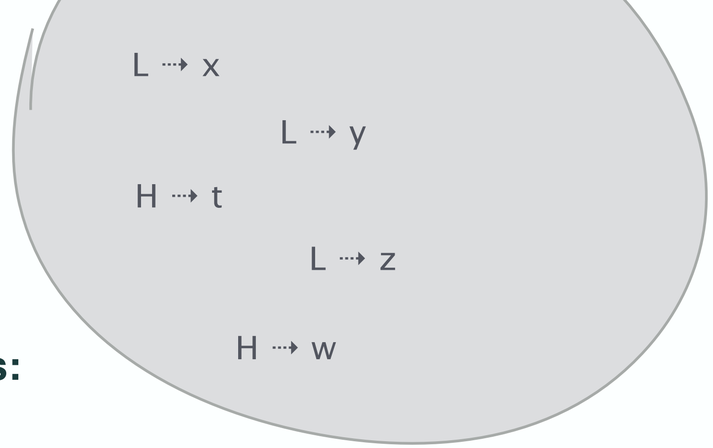
L ⇢ passing, H ⇢ english, math, science, bonus

L ⇢ passing, H ⇢ english, math, science, bonus

H ⇢ english, math, science, bonus, passing

H ⇢ english, math, science, bonus, passing

Secure Information Flow



$[\![P]\!]_{\mathrm{F}}$

possibilistic non-interference coincides with input data (non-)usage when the set J of unused input variables contains *all* input variables:

- **input variables are high-security variables**
- **output variables are low-security variables**

and the program is terminating

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

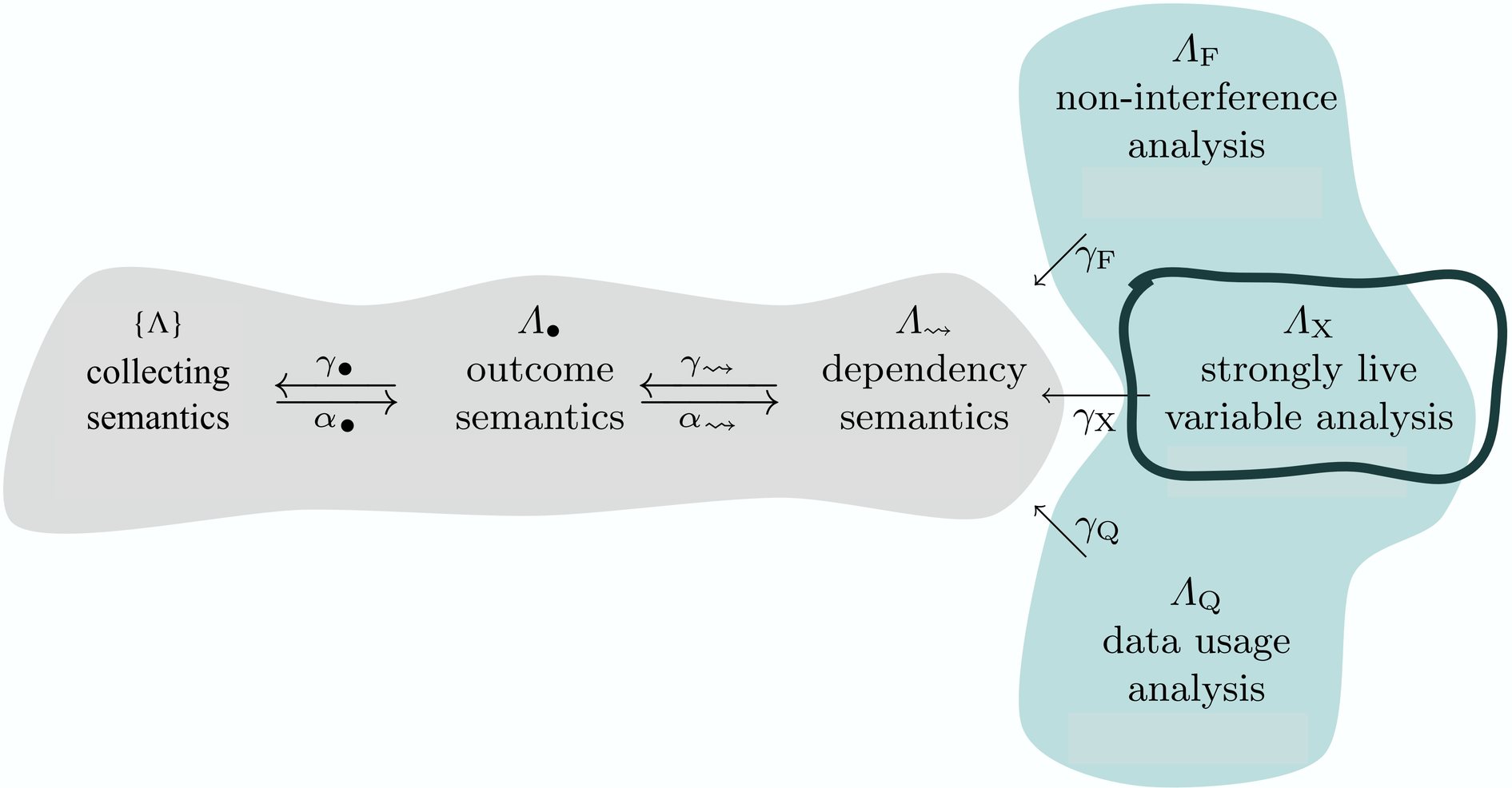
$$\Theta_{\mathrm{F}}[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$
$$\Theta_{\mathrm{F}}[\![x = e]\!](S) \stackrel{\text{def}}{=} \{L \rightsquigarrow y \in S \mid y \neq x\} \cup \{L \rightsquigarrow x \mid \mathcal{V}_{\mathrm{F}}[\![e]\!]S\}$$
$$\Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mathrm{F}}[\![s_1]\!](S) \sqcup_{\mathrm{F}} \Theta_{\mathrm{F}}[\![s_2]\!](S) & \text{if } \mathcal{V}_{\mathrm{F}}[\![e]\!]S \\ \{L \rightsquigarrow x \in S \mid x \notin \mathrm{w}(s_1) \cup \mathrm{w}(s_2)\} & \text{otherwise} \end{cases}$$
$$\Theta_{\mathrm{F}}[\![\texttt{while}\ e\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \mathrm{lfp}_S^{\sqsubseteq_{\mathrm{F}}}\ \Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]$$
$$\Theta_{\mathrm{F}}[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_{\mathrm{F}}[\![s_2]\!] \circ \Theta_{\mathrm{F}}[\![s_1]\!](S)$$

```
passing = True
while not english:
    english = False
```

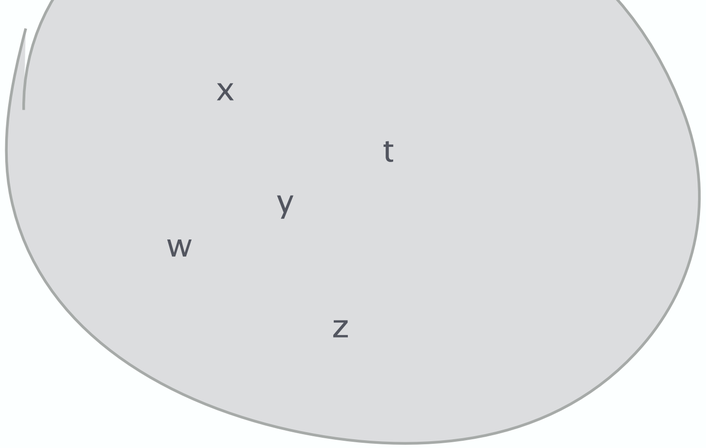
L ⇢ passing, H ⇢ english, math, science, bonus
L ⇢ passing, H ⇢ english, math, science, bonus
L ⇢ passing, H ⇢ english, math, science, bonus

Theorem

$$P \vDash \mathcal{N}_J^+ \Leftarrow \{[\![P]\!]\} \subseteq [\![P]\!]_F^{\natural} \subseteq \mathcal{N}_J^+$$


$\{\Lambda\}$
collecting semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome semantics
$\gamma_{\rightsquigarrow}$
$\alpha_{\rightsquigarrow}$
$\Lambda_{\rightsquigarrow}$
dependency semantics
Λ_{F}
non-interference analysis
γ_{F}
γ_{X}
Λ_{X}
strongly live variable analysis
γ_{Q}
Λ_{Q}
data usage analysis

Strong Liveness



$[\![P]\!]_X$

a variable is strongly live if

- **it is used in an assignment to another strongly live variable**
- **it is used in a statement other than an assignment**

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\Theta_X[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$

$$\Theta_X[\![x = e]\!](S) \stackrel{\text{def}}{=} \begin{cases} (S \setminus \{x\}) \cup \text{VARS}(e) & x \in S \\ S & \text{otherwise} \end{cases}$$

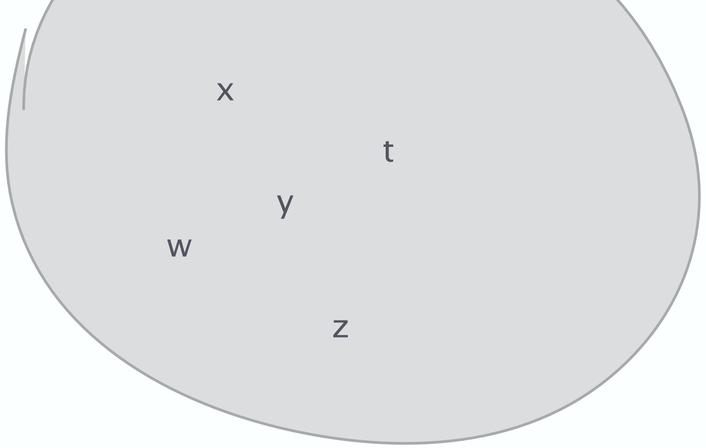
$$\Theta_X[\![\texttt{if}\ b\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \text{VARS}(b) \cup \Theta_X[\![s_1]\!](S) \cup \Theta_X[\![s_2]\!](S)$$

$$\Theta_X[\![\texttt{while}\ b\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \text{VARS}(b) \cup \Theta_X[\![s]\!](S)$$

$$\Theta_X[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_X[\![s_1]\!] \circ \Theta_X[\![s_2]\!](S)$$

$\langle \mathcal{P}(X), \subseteq, \cup, \cap, \emptyset, X \rangle$: abstract domain

Strong Liveness



$[\![P]\!]_X$

a variable is strongly live if

- **it is used in an assignment to another strongly live variable**
- **it is used in a statement other than an assignment**

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\Theta_X[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$

$$\Theta_X[\![x = e]\!](S) \stackrel{\text{def}}{=} \begin{cases} (S \setminus \{x\}) \cup \text{VARS}(e) & x \in S \\ S & \text{otherwise} \end{cases}$$

$$\Theta_X[\![\texttt{if}\ b\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \text{VARS}(b) \cup \Theta_X[\![s_1]\!](S) \cup \Theta_X[\![s_2]\!](S)$$

$$\Theta_X[\![\texttt{while}\ b\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \text{VARS}(b) \cup \Theta_X[\![s]\!](S)$$

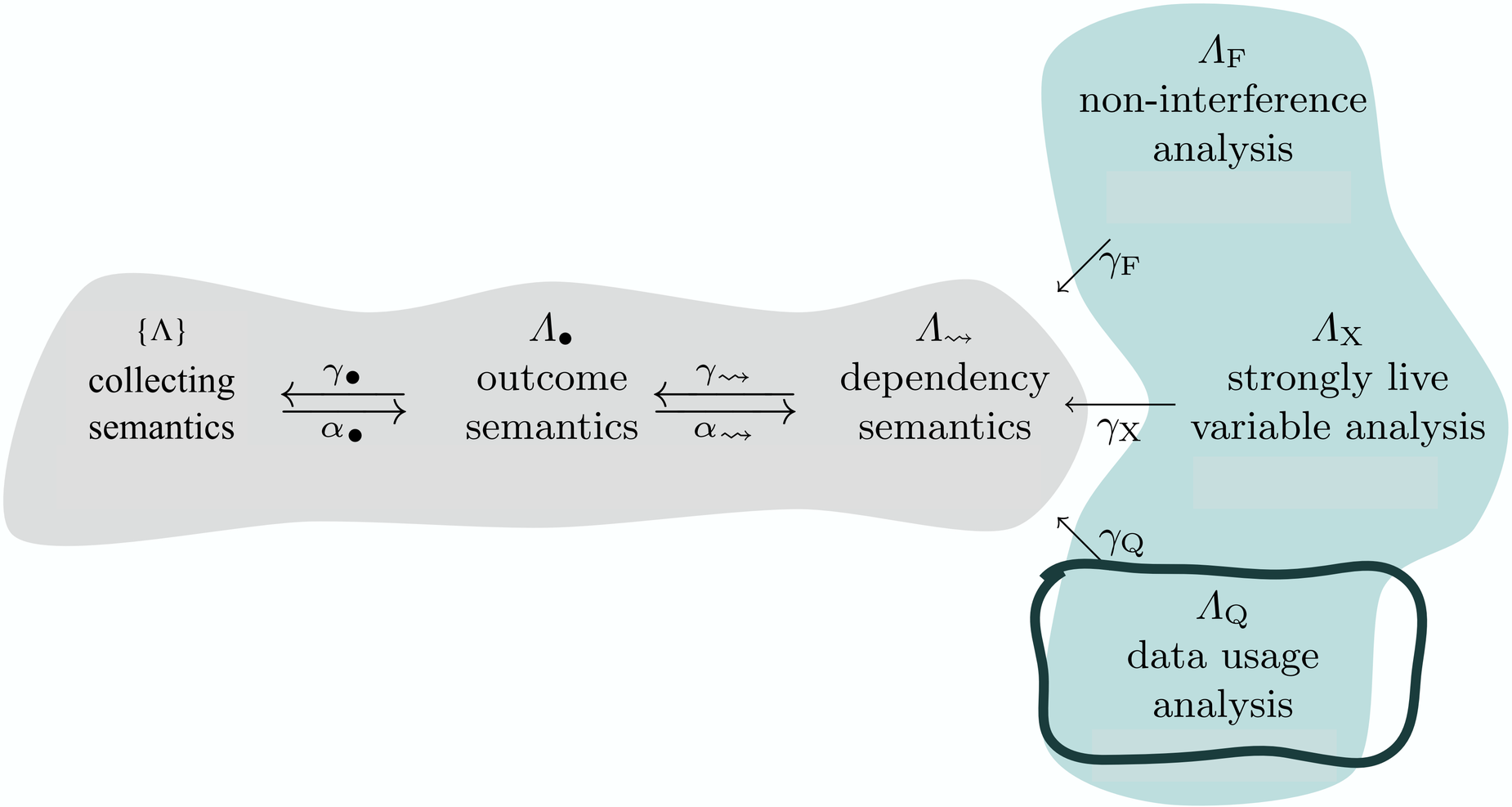
$$\Theta_X[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_X[\![s_1]\!] \circ \Theta_X[\![s_2]\!](S)$$

Theorem

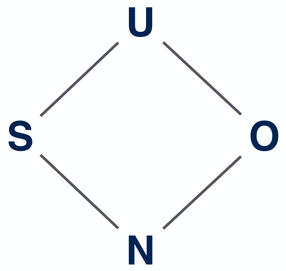
$$P \vDash \mathcal{N}_J \Leftarrow \{[\![P]\!]\} \subseteq [\![P]\!]^{\natural}_X \subseteq \mathcal{N}_J$$

```
passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

◀	{ bonus, math, english }
◀	{ bonus, math, english }
◀	{ bonus, math }
◀	{ bonus, math }
◀	{ bonus, math }
◀	{ bonus, math }
◀	{ bonus }
	{ passing }


Λ_{F}
non-interference
analysis
γ_{F}
$\{\Lambda\}$
collecting
semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome
semantics
$\gamma_{\rightsquigarrow}$
$\alpha_{\rightsquigarrow}$
$\Lambda_{\rightsquigarrow}$
dependency
semantics
γ_{X}
Λ_{X}
strongly live
variable analysis
γ_{Q}
Λ_{Q}
data usage
analysis

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

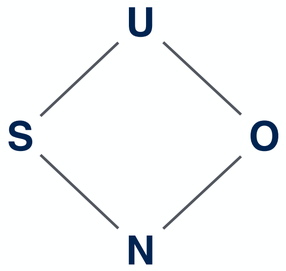
z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$
\begin{aligned}
\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) &\stackrel{\text{def}}{=} q \\
\Theta_{\mathrm{Q}}[\![x = e]\!](q) &\stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q) \\
\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) &\stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \\
&\quad \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q) \\
\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) &\stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!] \\
\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) &\stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)
\end{aligned}
$$

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

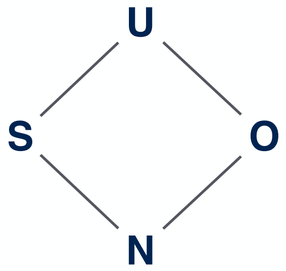
$[\![P]\!]_Q$

$$
\begin{aligned}
\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) &\stackrel{\text{def}}{=} q \\
\Theta_{\mathrm{Q}}[\![x = e]\!](q) &\stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q) \\
\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) &\stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \\
&\quad \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q) \\
\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) &\stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!] \\
\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) &\stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)
\end{aligned}
$$

```
passing = True
if not english:
    english = False

if not math:
    passing = False or bonus

if not math:
    passing = False or bonus
```

passing ⇢ U (any other variable maps to N)

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

```
passing = True

if not english:

    english = False

if not math:

    passing = False or bonus

if not math:

    passing = False or bonus
```

passing ⇢ S | passing ⇢ U

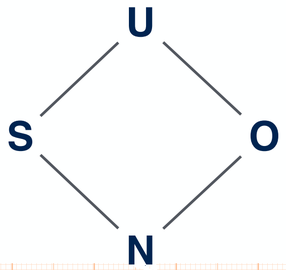
passing ⇢ U

x → U ⇒ x → S

x → O ⇒ x → N

domain elements are stacks of maps matching nesting level of analyzed statements

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

if the assigned variable was used (U or S) it becomes overwritten (O) if not also used in the expression being assigned; otherwise it becomes freshly used (U)

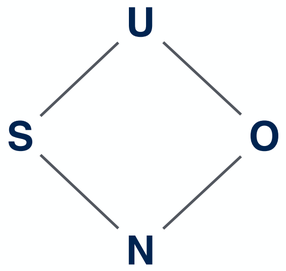
```
passing = True

if not english:

    english = False


if not math:

    passing = False or bonus


if not math:
    passing = False or bonus
```

bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}} \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

```
passing = True
if not english:
    english = False

if not math:
    passing = False or bonus

if not math:
    passing = False or bonus
```

math, bonus, passing ⇢ U

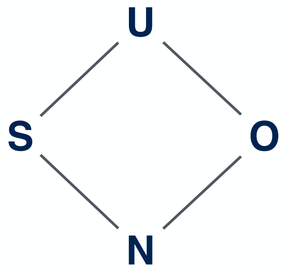
bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

a variable becomes used (U) if it appears in the boolean condition of a statement that uses (U) or modifies (O) another variable

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

restores the previous value of variables (before increasing the nesting level) if it has not changed since

```
passing = True
if not english:
    english = False

if not math:
    passing = False or bonus

if not math:
    passing = False or bonus
```

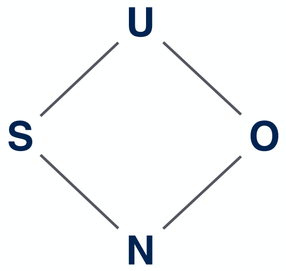
math, bonus, passing ⇢ U
math, bonus ↦ U, passing ↦ O

bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_{\mathrm{t}}^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

```
passing = True
if not english:
    english = False

if not math:
    passing = False or bonus

if not math:
    passing = False or bonus
```

math, bonus, passing ⇢ U

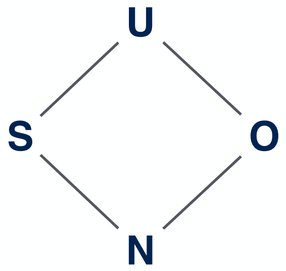
bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

math, bonus ↦ U, passing ↦ O ⊔Q passing ↦ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$
\begin{aligned}
\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) &\stackrel{\text{def}}{=} q \\
\Theta_{\mathrm{Q}}[\![x = e]\!](q) &\stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q) \\
\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) &\stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \\
&\quad \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q) \\
\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) &\stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!] \\
\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) &\stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)
\end{aligned}
$$

```
passing = True
if not english:
    english = False

if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

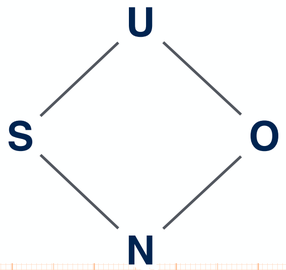
math, bonus, passing ⇢ U

bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

```
passing = True
if not english:
    english = False

if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

math ⇢ S, bonus ⇢ U, passing ⇢ O | …

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

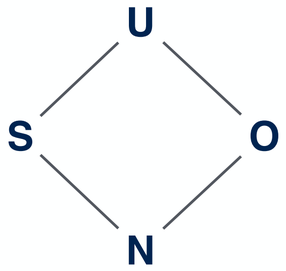
math, bonus, passing ⇢ U

bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_Q[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_Q[\![x = e]\!](q) \stackrel{\text{def}}{=} \text{ASSIGN}[\![x = e]\!](q)$$

$$\Theta_Q[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_1]\!] \circ \text{PUSH}(q) \sqcup_Q \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_2]\!] \circ \text{PUSH}(q)$$

$$\Theta_Q[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \text{lfp}_{\bot}^{\sqsubseteq_Q} \Theta_Q[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_Q[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_Q[\![s_1]\!] \circ \Theta_Q[\![s_2]\!](q)$$

```
passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

math, bonus, passing ⇢ U

math ⇢ S, bonus ⇢ U, passing ⇢ O | …

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

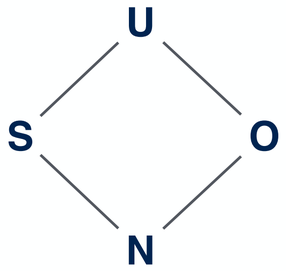
math, bonus, passing ⇢ U

bonus ⇢ U, passing ⇢ O | passing ⇢ U

passing ⇢ S | passing ⇢ U

passing ⇢ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \textsc{assign}[\![x = e]\!](q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \textsc{push}(q) \sqcup_{\mathrm{Q}} \textsc{pop} \circ \textsc{filter}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \textsc{push}(q)$$

$$\Theta_{\mathrm{Q}}[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

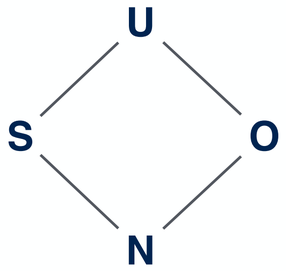
$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

```
passing = True
if not english:
    english = False
```
math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
math, bonus, passing ⇢ U

```
if not math:
    passing = False or bonus
```
math ⇢ S, bonus ⇢ U, passing ⇢ O | …
math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
math, bonus, passing ⇢ U

```
if not math:
    passing = False or bonus
```
bonus ⇢ U, passing ⇢ O | passing ⇢ U
passing ⇢ S | passing ⇢ U
passing ⇢ U

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

```
passing = True

if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

math, bonus, passing ⇢ U

passing ⇢ U

$$\Theta_Q[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

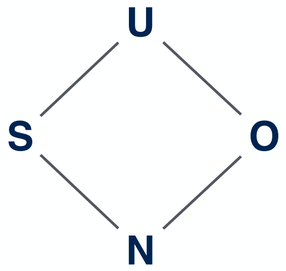
$$\Theta_Q[\![x = e]\!](q) \stackrel{\text{def}}{=} \text{ASSIGN}[\![x = e]\!](q)$$

$$\Theta_Q[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_1]\!] \circ \text{PUSH}(q) \sqcup_Q \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_2]\!] \circ \text{PUSH}(q)$$

$$\Theta_Q[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \text{lfp}_t^{\sqsubseteq_Q}\, \Theta_Q[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_Q[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_Q[\![s_1]\!] \circ \Theta_Q[\![s_2]\!](q)$$

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

```
passing = True
                                    math, bonus, passing ⇢ U
if not english:
    english = False                 math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
                                    math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
if not math:                        math, bonus, passing ⇢ U
    passing = False or bonus
if not math:
    passing = False or bonus
                                    passing ⇢ U
```

$$\Theta_Q[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_Q[\![x = e]\!](q) \stackrel{\text{def}}{=} \text{ASSIGN}[\![x = e]\!](q)$$

$$\Theta_Q[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_1]\!] \circ \text{PUSH}(q) \sqcup_Q \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_2]\!] \circ \text{PUSH}(q)$$

$$\Theta_Q[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \text{lfp}_t^{\sqsubseteq_Q}\, \Theta_Q[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_Q[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_Q[\![s_1]\!] \circ \Theta_Q[\![s_2]\!](q)$$

Syntactic (Non-)Usage

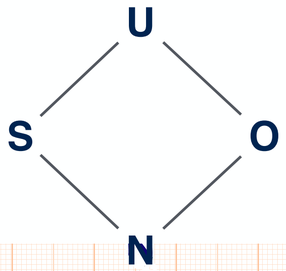
x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

$[\![P]\!]_Q$

the input variables english and science are definetly not used by the program

```
                                                  math, bonus ⇢ U, passing ⇢ O
passing = True
                                                  math, bonus, passing ⇢ U
if not english:
                                                  math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
    english = False
                                                  math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
                                                  math, bonus, passing ⇢ U
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
                                                  passing ⇢ U
```

$$\Theta_Q[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

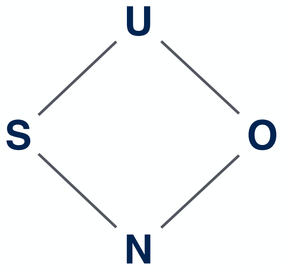
$$\Theta_Q[\![x = e]\!](q) \stackrel{\text{def}}{=} \text{ASSIGN}[\![x = e]\!](q)$$

$$\Theta_Q[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_1]\!] \circ \text{PUSH}(q) \sqcup_Q \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_2]\!] \circ \text{PUSH}(q)$$

$$\Theta_Q[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \text{lfp}_t^{\sqsubseteq_Q}\, \Theta_Q[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_Q[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_Q[\![s_1]\!] \circ \Theta_Q[\![s_2]\!](q)$$

Syntactic (Non-)Usage



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

x ⇢ U

y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$
$$\Theta_{\mathrm{Q}}[\![x = e]\!](q) \stackrel{\text{def}}{=} \mathrm{ASSIGN}[\![x = e]\!](q)$$
$$\Theta_{\mathrm{Q}}[\![\texttt{if}\ b\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](q) \stackrel{\text{def}}{=} \mathrm{POP} \circ \mathrm{FILTER}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \mathrm{PUSH}(q) \sqcup_{\mathrm{Q}} \mathrm{POP} \circ \mathrm{FILTER}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \mathrm{PUSH}(q)$$
$$\Theta_{\mathrm{Q}}[\![\texttt{while}\ b\texttt{:}\ s]\!](q) \stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if}\ b\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]$$
$$\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)$$

```
passing = True
while not english:
    english = False
```

passing ⇢ O

passing ⇢ U

passing ⇢ U

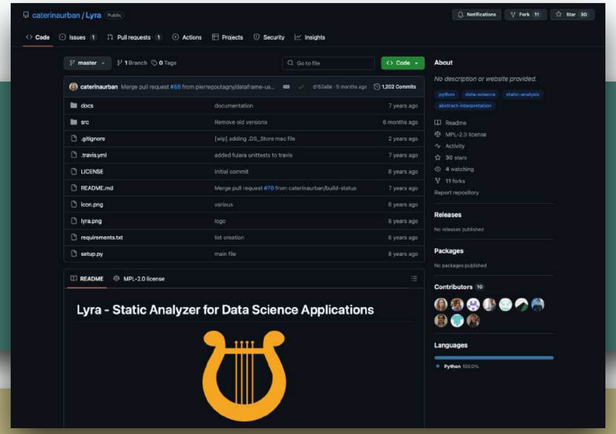
Theorem

$$P \vDash \mathcal{N}_J^+ \Leftarrow \{[\![P]\!]\} \subseteq [\![P]\!]_Q^\natural \subseteq \mathcal{N}_J^+$$

Data Usage Static Analysis [Urban18]

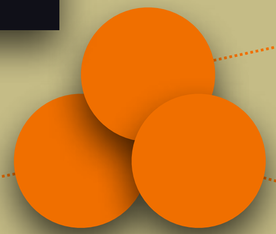
3-Step Recipe

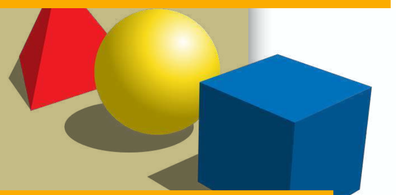
practical tools
targeting specific programs



abstract semantics, abstract domains
algorithmic approaches to decide program properties

secure information flow

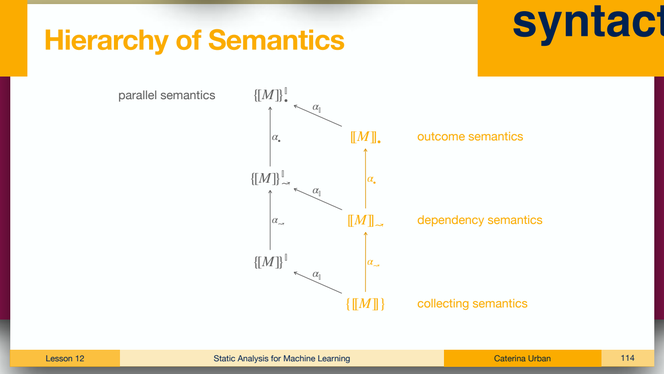




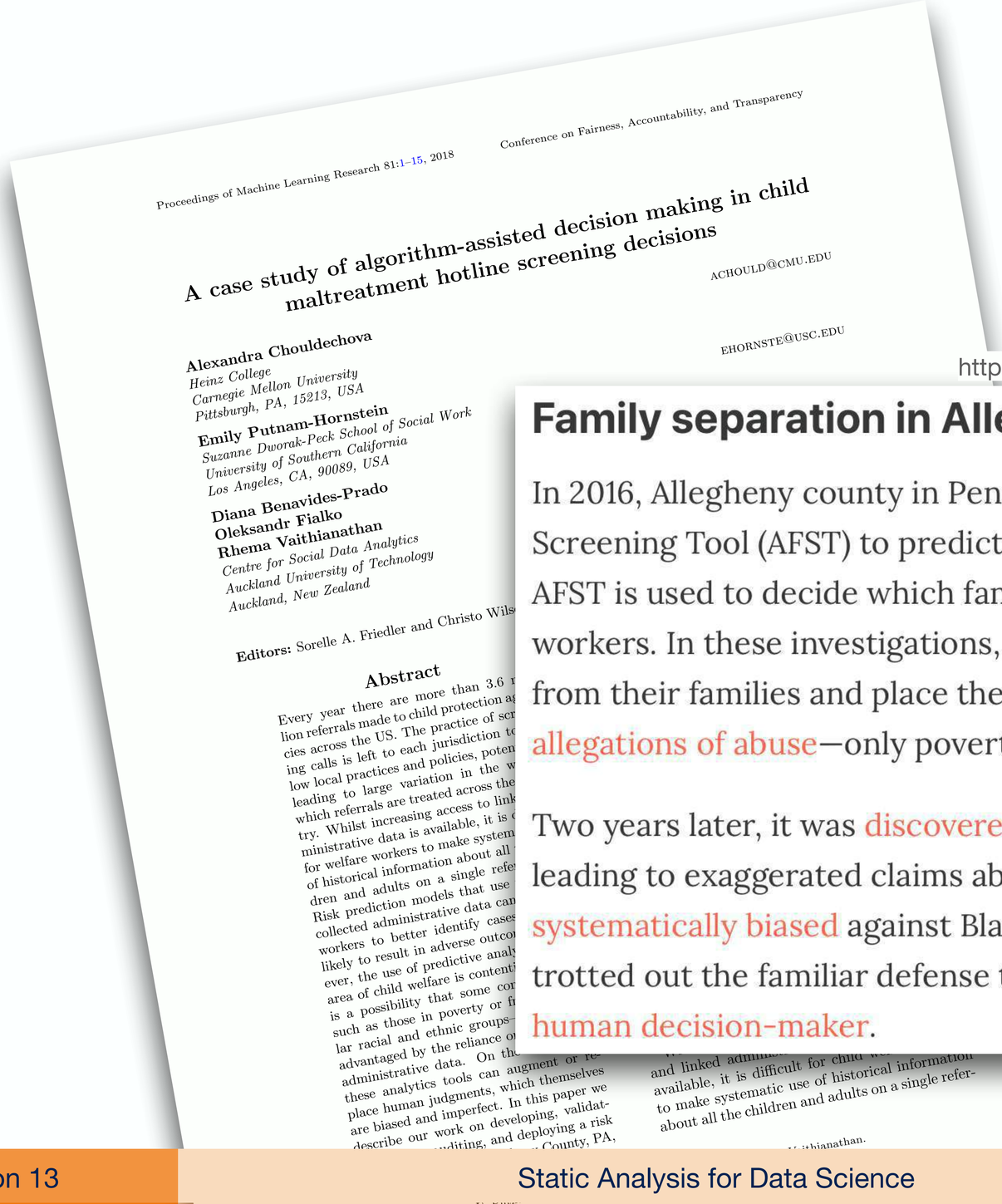
strongly-live variable analysis

syntactic non-usage

concrete semantics
mathematical models of the program behavior



Data Leakage



Proceedings of Machine Learning Research 81:1–15, 2018 Conference on Fairness, Accountability, and Transparency

A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions

Alexandra Chouldechova ACHOULD@CMU.EDU
Heinz College
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

Emily Putnam-Hornstein EHORNSTE@USC.EDU
Suzanne Dworak-Peck School of Social Work
University of Southern California
Los Angeles, CA, 90089, USA

Diana Benavides-Prado
Oleksandr Fialko
Rhema Vaithianathan
Centre for Social Data Analytics
Auckland University of Technology
Auckland, New Zealand

Editors: Sorelle A. Friedler and Christo Wils

Abstract

Every year there are more than 3.6 m
lion referrals made to child protection ag
cies across the US. The practice of scr
ing calls is left to each jurisdiction to
low local practices and policies, poten
leading to large variation in the w
which referrals are treated across the
try. Whilst increasing access to link
ministrative data is available, it is c
for welfare workers to make system
of historical information about all
dren and adults on a single refe
Risk prediction models that use
collected administrative data can
workers to better identify cases
likely to result in adverse outcom
ever, the use of predictive analy
area of child welfare is contenti
is a possibility that some com
such as those in poverty or fa
lar racial and ethnic groups—
advantaged by the reliance o
administrative data. On th
these analytics tools can augment or re-
place human judgments, which themselves
are biased and imperfect. In this paper we
describe our work on developing, validat-
uditing, and deploying a risk
County, PA,
and linked admini
available, it is difficult for child w
to make systematic use of historical information
about all the children and adults on a single refer-
Vaithianathan.

https://www.aisnakeoil.com/p/the-bait-and-switch-behind-ai-risk

Family separation in Allegheny county

In 2016, Allegheny county in Pennsylvania adopted the Allegheny Family Screening Tool (AFST) to predict which children are at risk of maltreatment. AFST is used to decide which families should be investigated by social workers. In these investigations, social workers can forcibly remove children from their families and place them in foster care, even if there are no allegations of abuse—only poverty-based neglect.

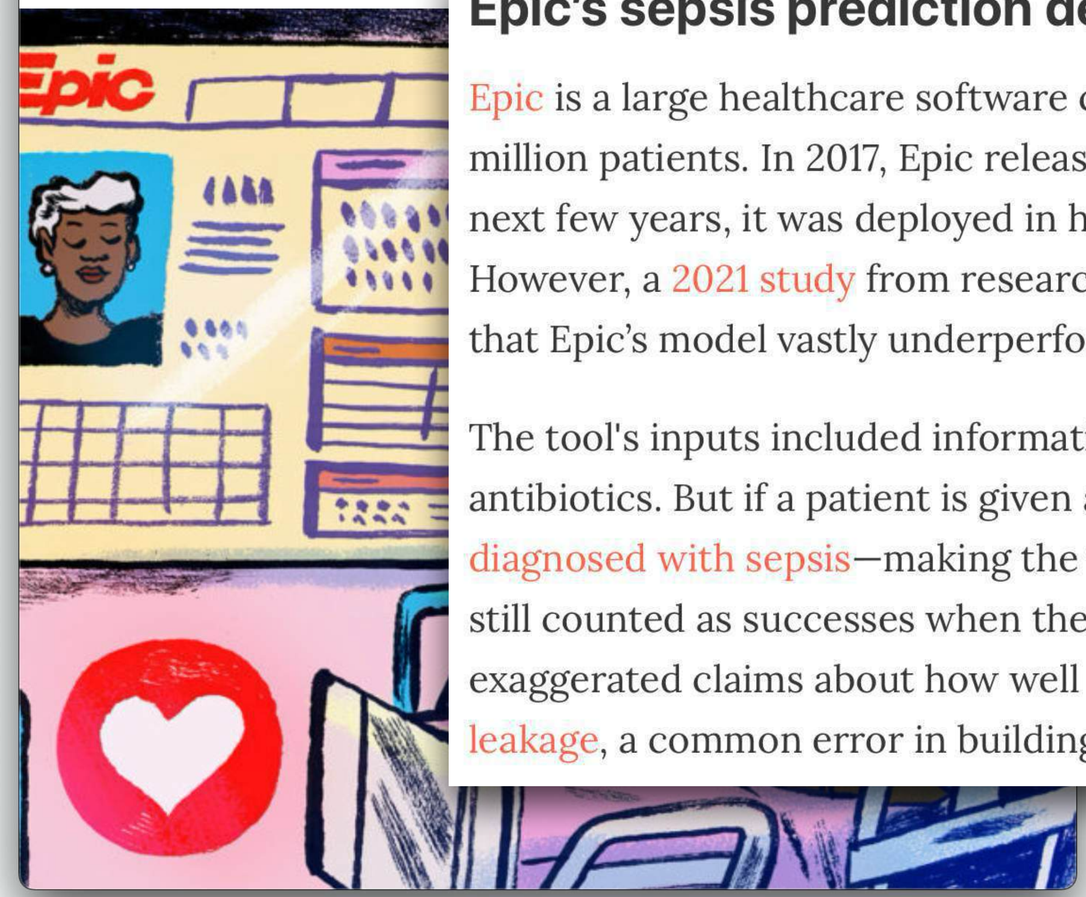
Two years later, it was discovered that AFST suffered from data leakage, leading to exaggerated claims about its performance. In addition, the tool was systematically biased against Black families. When questioned, the creators trotted out the familiar defense that the final decision is always made by a human decision-maker.

statnews.com

A STAT INVESTIGATION

Epic's sepsis algorithm is going off the rails in the real world. The use of these variables may explain why

By Casey Ross Sept. 27, 2021



https://www.aisnakeoil.com/p/the-bait-and-switch-behind-ai-risk

Epic's sepsis prediction debacle

Epic is a large healthcare software company. It stores health data for over 300 million patients. In 2017, Epic released a sepsis prediction model. Over the next few years, it was deployed in hundreds of hospitals across the U.S. However, a 2021 study from researchers at the University of Michigan found that Epic's model vastly underperformed compared to the developer's claims.

The tool's inputs included information about whether a patient was given antibiotics. But if a patient is given antibiotics, they have already been diagnosed with sepsis—making the tool's prediction useless. These cases were still counted as successes when the developer evaluated the tool, leading to exaggerated claims about how well it performed. This is an example of data leakage, a common error in building AI tools.

Example

```
[1]: import numpy as np
     import pandas as pd
     from sklearn.preprocessing import MinMaxScaler
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import accuracy_score

[2]: data = pd.read_csv("data.csv")
     X = data[["X_1", "X_2"]]
     y = data[["y"]]

[3]: min_max_scaler = MinMaxScaler()
     X = min_max_scaler.fit_transform(X)

[4]: X_train , X_test , y_train , y_test = train_test_split(X, y, test_size=0.025, random_state=2)

[ ]:

[5]: lr = LogisticRegression()
     a = lr.fit(X_train , y_train)

[6]: y_pred = lr.predict(X_test)
     accuracy_score(y_test , y_pred)

[6]: 0.67
```

INPUT DATA READING

MIN-MAX NORMALIZATION

TRAIN/TEST SPLIT

TRAINING

TESTING

Example

```
[1]: import numpy as np
     import pandas as pd
     from sklearn.preprocessing import MinMaxScaler
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import accuracy_score

[2]: data = pd.read_csv("data.csv")
     X = data[["X_1", "X_2"]]
     y = data[["y"]]

[ ]:

[3]: X_train , X_test , y_train , y_test = train_test_split(X, y, test_size=0.025, random_state=2)

[4]: min_max_scaler = MinMaxScaler()
     X_train = min_max_scaler.fit_transform(X_train)
     X_test = min_max_scaler.fit_transform(X_test)

[5]: lr = LogisticRegression()
     a = lr.fit(X_train , y_train)

[6]: y_pred = lr.predict(X_test)
     accuracy_score(y_test , y_pred)

[6]: 0.33
```

INPUT DATA READING

TRAIN/TEST SPLIT

MIN-MAX NORMALIZATION

TRAINING

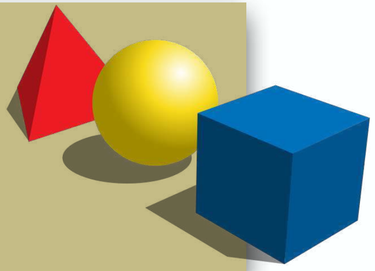
TESTING

Data Leakage Static Analysis [Drobnjaković24]

3-Step Recipe

practical tools
targeting specific programs

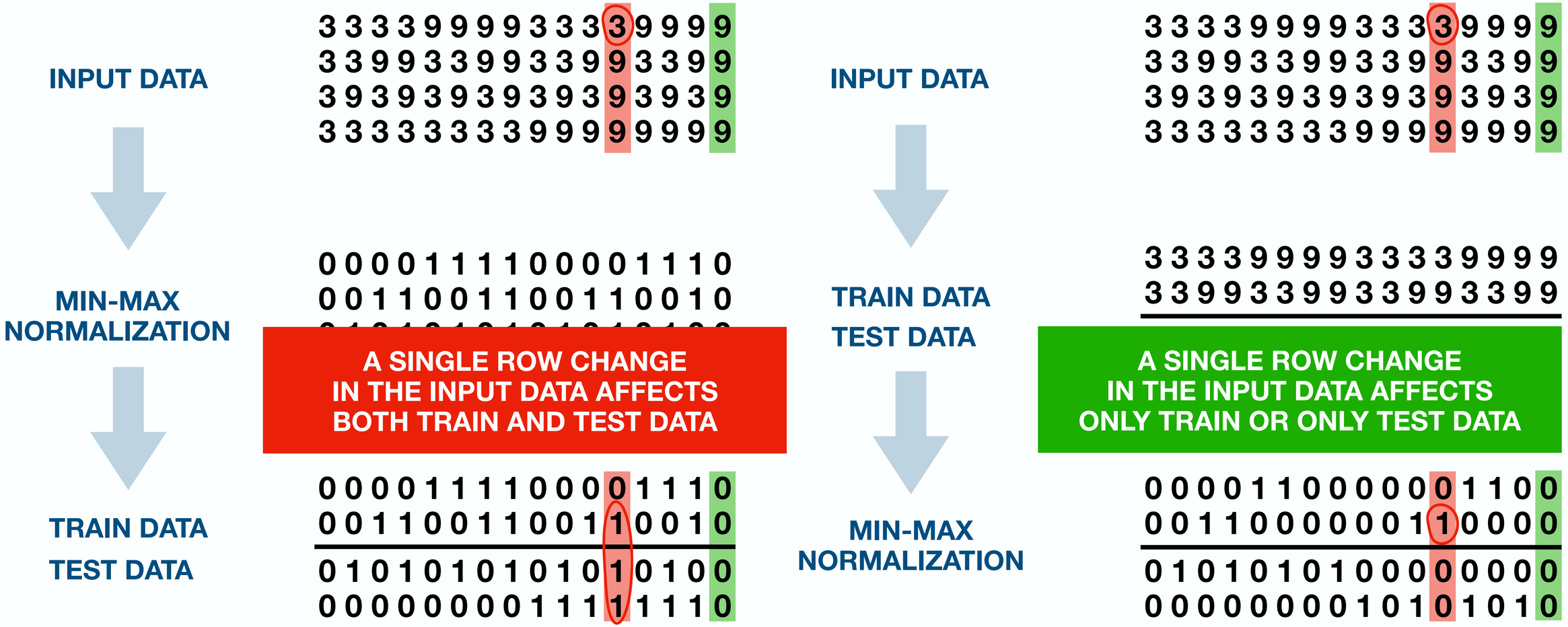
abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

(Absence of) Data Leakage

Independence of Training and Testing Data



(Absence of) Data Leakage

Independence of Training and Testing Data

$$\mathscr{I} \overset{\text{def}}{=} \{ [\![P]\!] \mid \text{INDEPENDENT}([\![P]\!]) \}$$

input data frame variables — data frame row — used data frame variables

$$\text{INDEPENDENT}([\![P]\!]) \overset{\text{def}}{=} \forall \sigma \in [\![P]\!], i \in \mathrm{I}_P, r \in R_i : \text{UNCHANGED}(\sigma, i, r, \mathrm{U}_P^{\text{test}}) \vee \text{UNCHANGED}(\sigma, i, r, \mathrm{U}_P^{\text{train}})$$

$$\text{UNCHANGED}(\sigma, i, r, U) \overset{\text{def}}{=} \forall \bar{v} \in \mathbb{V}^{C_i} : \sigma(i)[r] \neq \bar{v} \Rightarrow (\exists \sigma' \in \Upsilon[\![P]\!] : \sigma'(i)[r] = \bar{v} \wedge \eta(\sigma) = \eta(\sigma') \wedge \sigma(U) = \sigma'(U))$$

$$\eta(\sigma) \overset{\text{def}}{=} \lambda j : \lambda r' : \begin{cases} \sigma(j)[r'] & j \in \mathrm{I}_P \setminus \{i\} \vee r' \in R_i : r' \neq r \\ \top & \text{otherwise} \end{cases}$$

$$\sigma(X) = \sigma'(X) \overset{\text{def}}{=} \forall x \in X : \sigma(x) = \sigma'(x)$$

Data Leakage Static Analysis [Drobnjaković24]

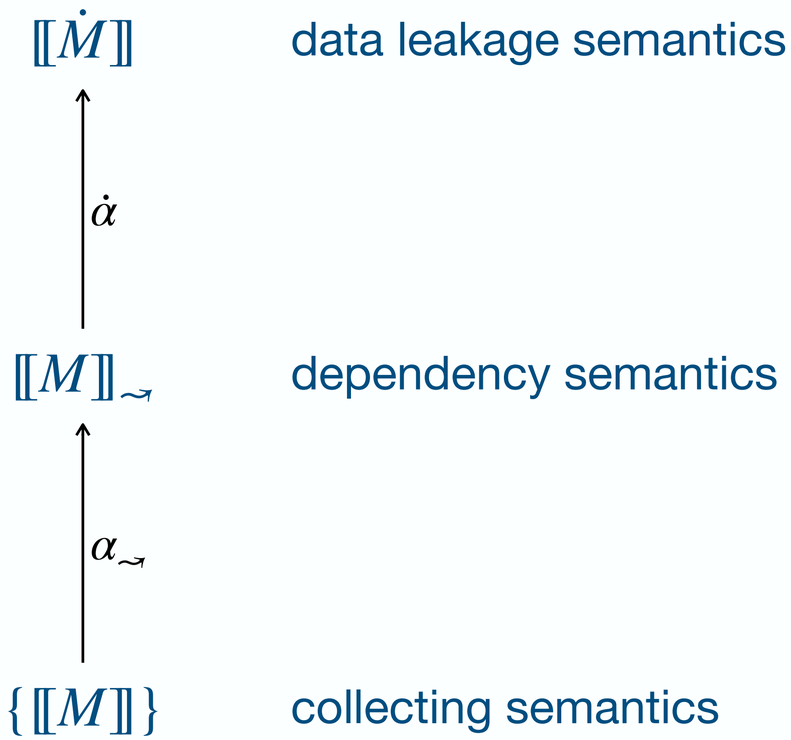
3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

concrete semantics
mathematical models of the program behavior

Hierarchy of Semantics



Hierarchy of Semantics

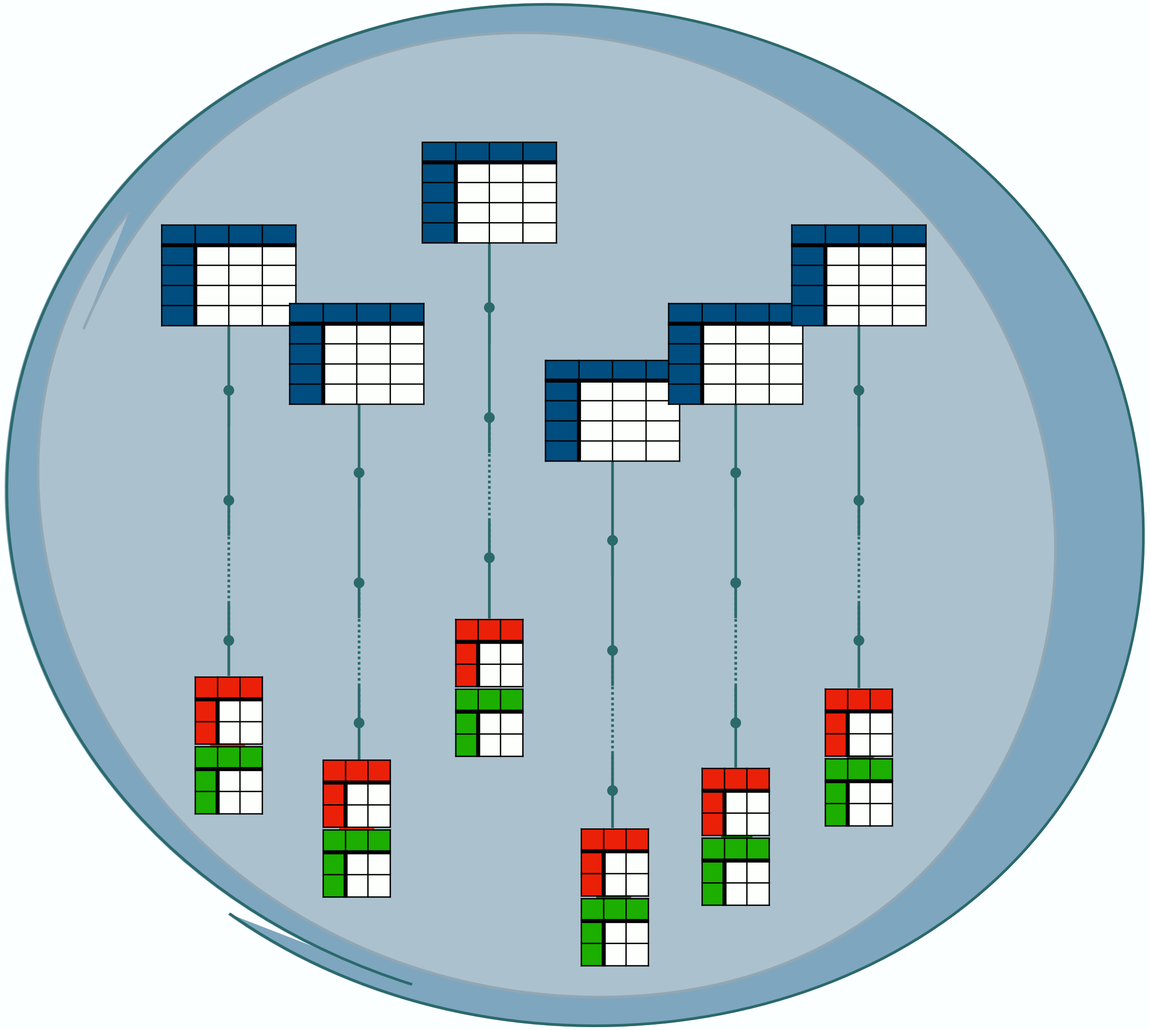
$[\![\dot{M}]\!]$ data leakage semantics

$\dot{\alpha}$

$[\![M]\!]_{\rightsquigarrow}$ dependency semantics

$\alpha_{\rightsquigarrow}$

$\{[\![M]\!]\}$ collecting semantics



Hierarchy of Semantics

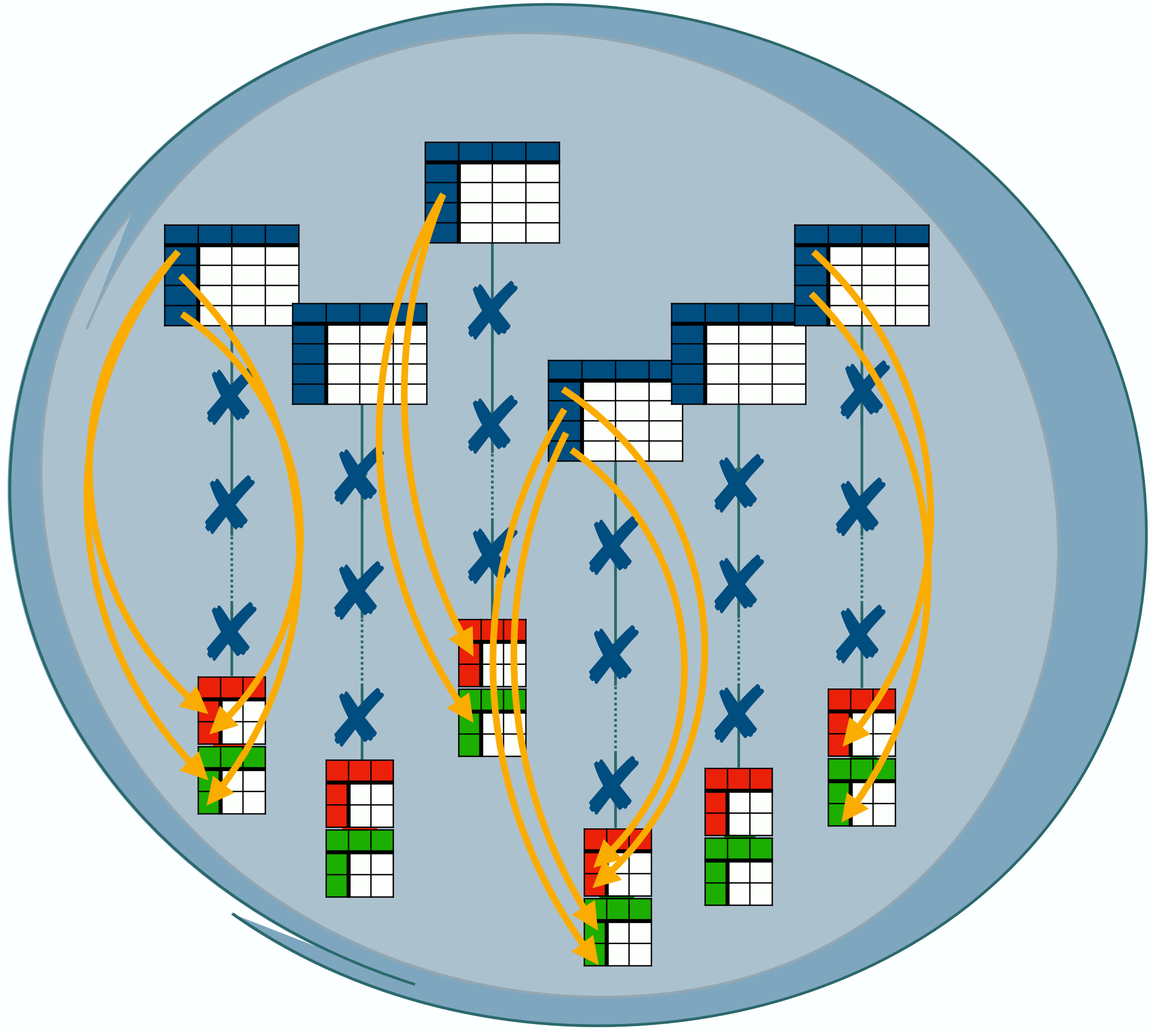
$[\![\dot{M}]\!]$ data leakage semantics

$\dot{\alpha}$

$[\![M]\!]_{\rightsquigarrow}$ dependency semantics

$\alpha_{\rightsquigarrow}$

$\{[\![M]\!]\}$ collecting semantics



Hierarchy of Semantics

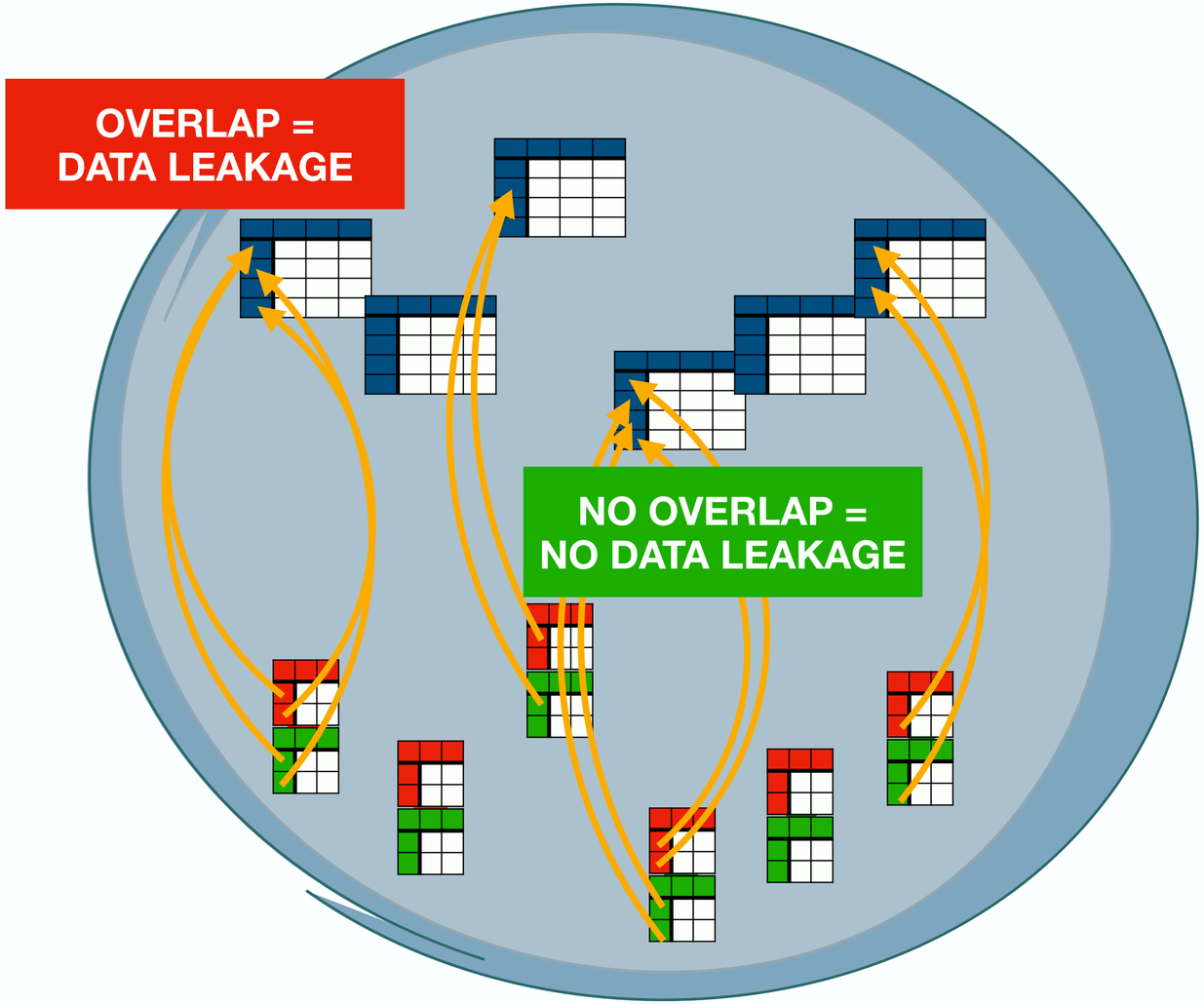
$[\![\dot{M}]\!]$ data leakage semantics

$\dot{\alpha}$

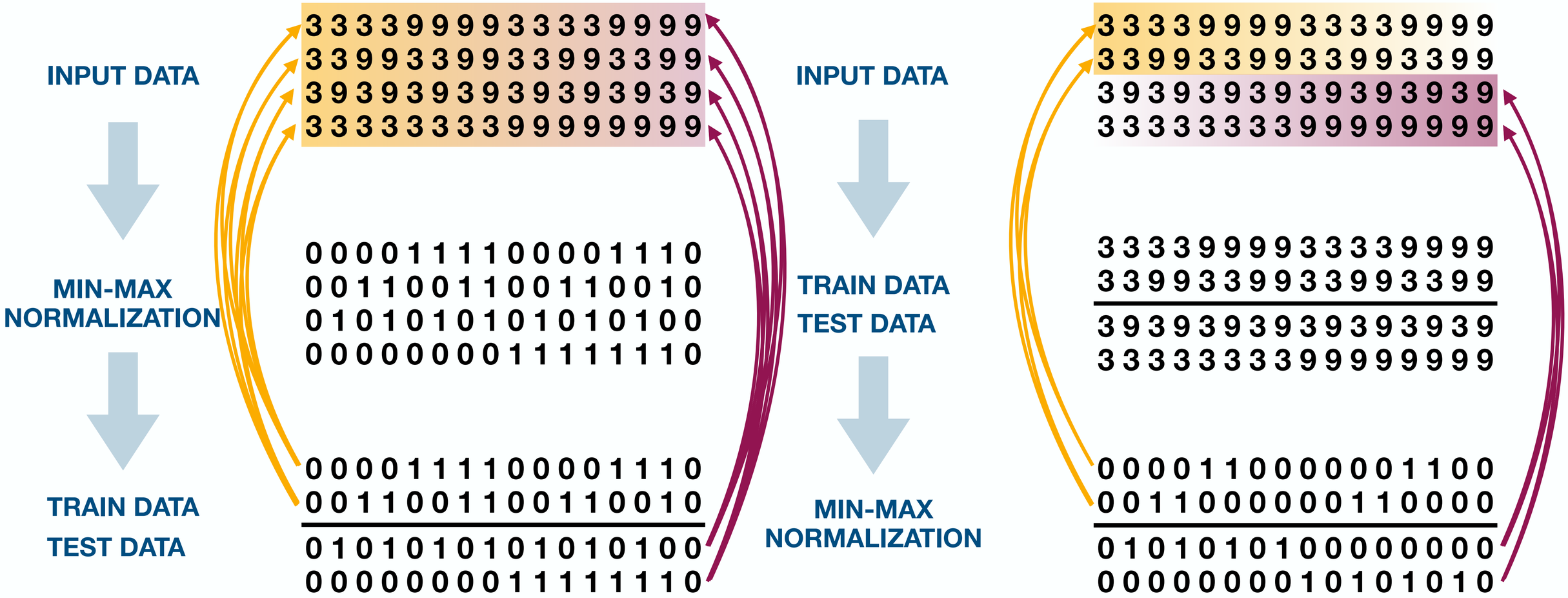
$[\![M]\!]_{\rightsquigarrow}$ dependency semantics

$\alpha_{\rightsquigarrow}$

$\{[\![M]\!]\}$ collecting semantics



Data Leakage Semantics

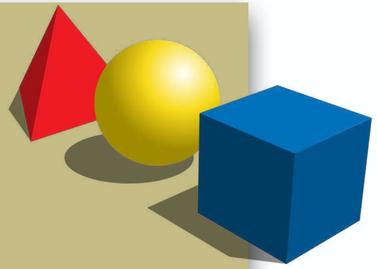


Data Leakage Static Analysis [Drobnjaković24]

3-Step Recipe

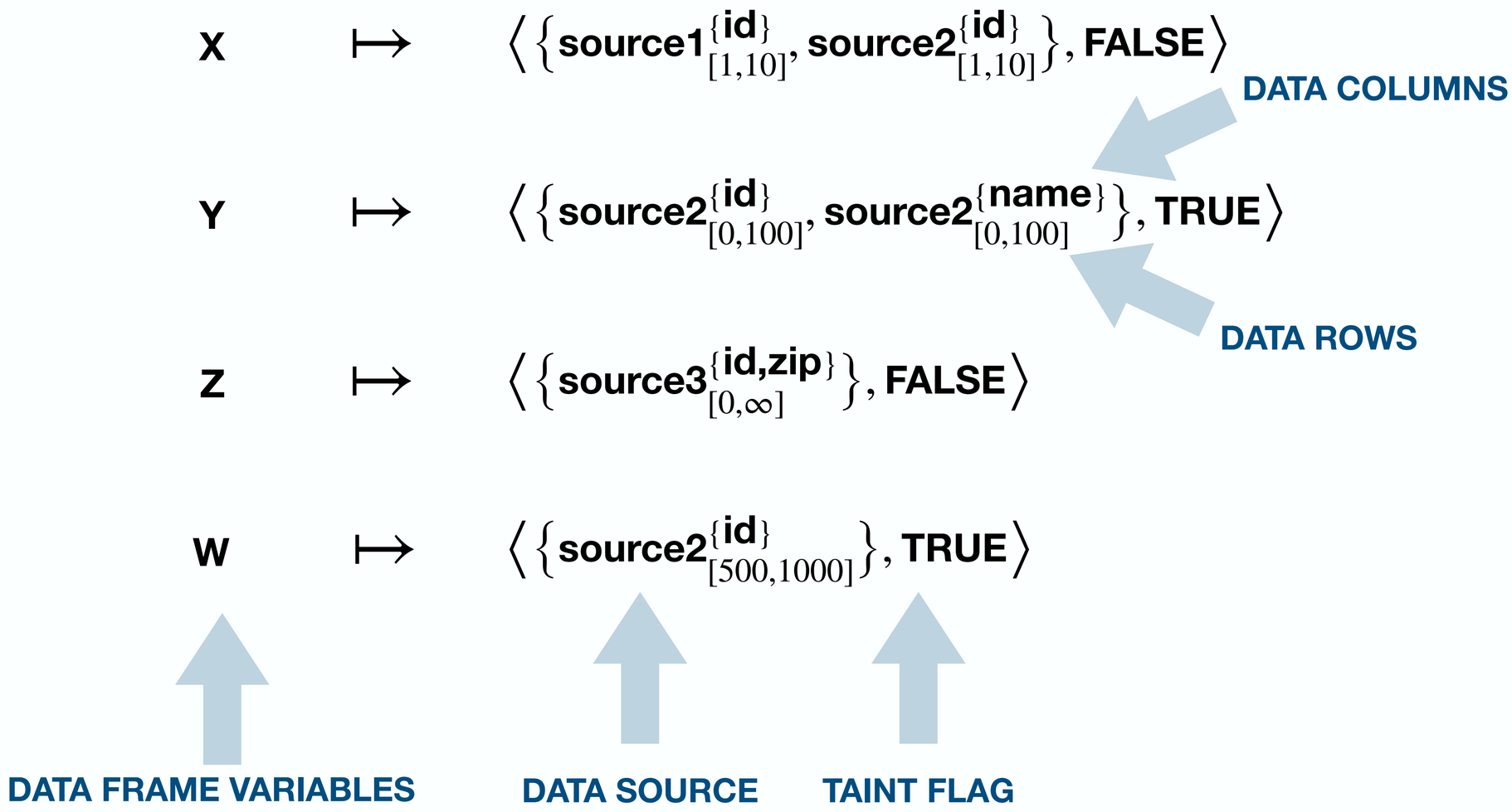
practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Data Frame Sources Abstract Domain



Example

```
[1]: import numpy as np
     import pandas as pd
     from sklearn.preprocessing import MinMaxScaler
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import accuracy_score

[2]: data = pd.read_csv("data.csv")
     X = data[["X_1", "X_2"]]
     y = data[["y"]]

[3]: min_max_scaler = MinMaxScaler()
     X = min_max_scaler.fit_transform(X)

[4]: X_train , X_test , y_train , y_test = train_test_split(X, y, test_size=0.025, random_state=2)

[ ]:

[5]: lr = LogisticRegression()
     a = lr.fit(X_train , y_train)

[6]: y_pred = lr.predict(X_test)
     accuracy_score(y_test , y_pred)

[6]: 0.67
```

INPUT DATA READING

$$X \mapsto \left\langle \left\{ \text{data}_{[0,\infty]}^{\{X_1,X_2\}} \right\}, \text{FALSE} \right\rangle$$

MIN-MAX NORMALIZATION

$$X \mapsto \left\langle \left\{ \text{data}_{[0,\infty]}^{\{X_1,X_2\}} \right\}, \text{TRUE} \right\rangle$$

TRAIN/TEST SPLIT

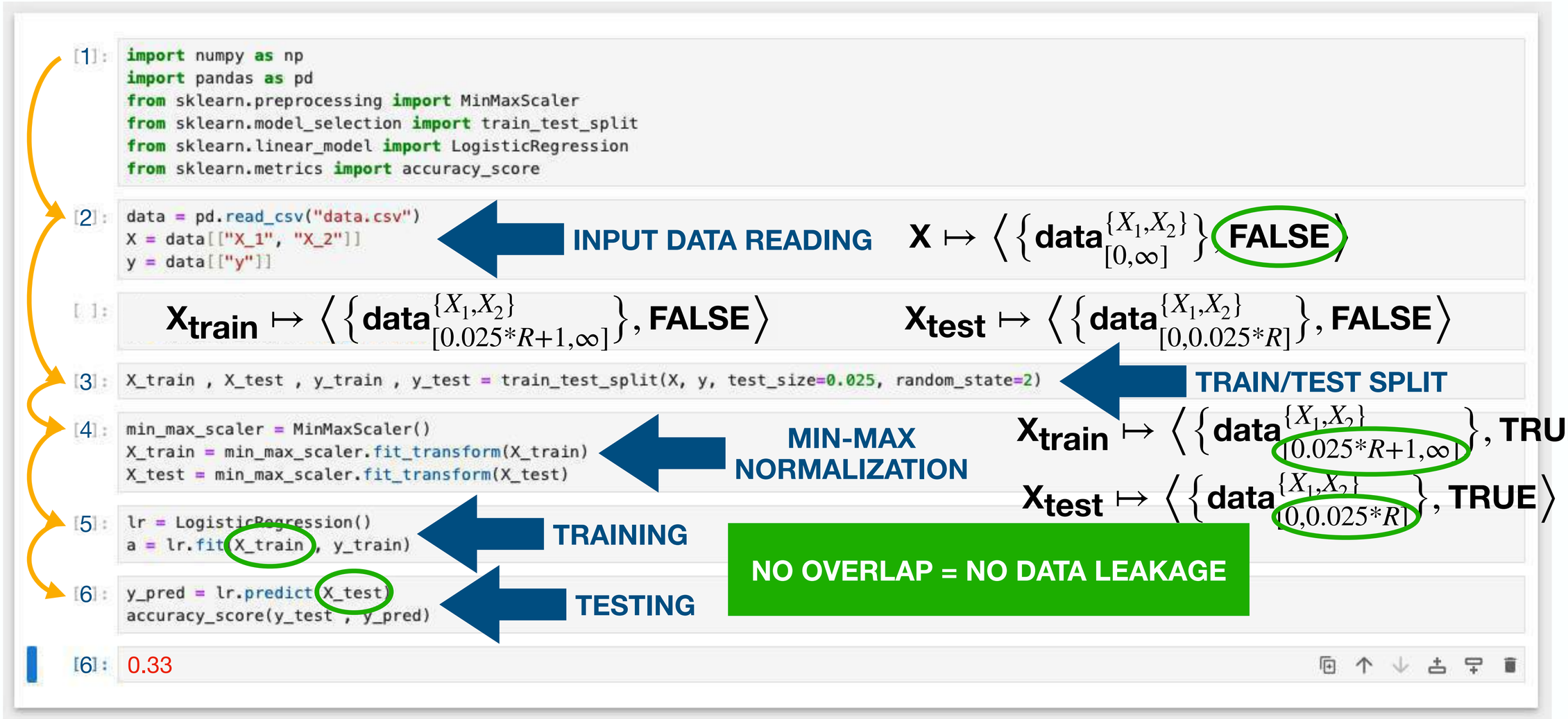
$$X_{\text{train}} \mapsto \left\langle \left\{ \text{data}_{[0,\infty]}^{\{X_1,X_2\}} \right\}, \text{TRUE} \right\rangle \qquad X_{\text{test}} \mapsto \left\langle \left\{ \text{data}_{[0,\infty]}^{\{X_1,X_2\}} \right\}, \text{TRUE} \right\rangle$$

TRAINING

TESTING

(TAINT) OVERLAP = DATA LEAKAGE

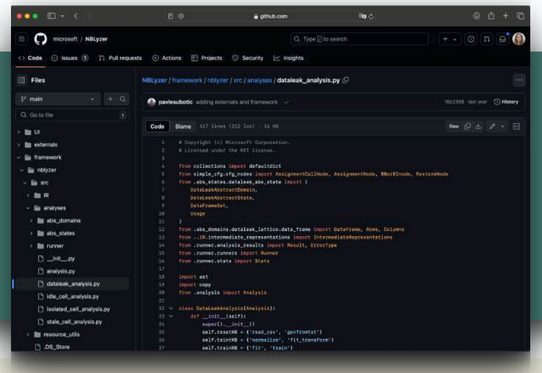
Example


[1]:
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
[2]:
data = pd.read_csv("data.csv")
X = data[["X_1", "X_2"]]
y = data[["y"]]
INPUT DATA READING
$X \mapsto \langle \{ data^{\{X_1,X_2\}}_{[0,\infty]} \}, FALSE \rangle$
[ ]:
$X_{train} \mapsto \langle \{ data^{\{X_1,X_2\}}_{[0.025*R+1,\infty]} \}, FALSE \rangle$
$X_{test} \mapsto \langle \{ data^{\{X_1,X_2\}}_{[0,0.025*R]} \}, FALSE \rangle$
[3]:
X_train , X_test , y_train , y_test = train_test_split(X, y, test_size=0.025, random_state=2)
TRAIN/TEST SPLIT
[4]:
min_max_scaler = MinMaxScaler()
X_train = min_max_scaler.fit_transform(X_train)
X_test = min_max_scaler.fit_transform(X_test)
MIN-MAX NORMALIZATION
$X_{train} \mapsto \langle \{ data^{\{X_1,X_2\}}_{[0.025*R+1,\infty]} \}, TRUE \rangle$
$X_{test} \mapsto \langle \{ data^{\{X_1,X_2\}}_{[0,0.025*R]} \}, TRUE \rangle$
[5]:
lr = LogisticRegression()
a = lr.fit(X_train, y_train)
TRAINING
[6]:
y_pred = lr.predict(X_test)
accuracy_score(y_test, y_pred)
TESTING
NO OVERLAP = NO DATA LEAKAGE
[6]:
0.33

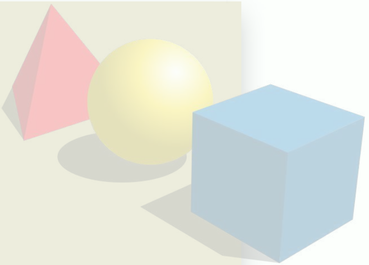
Data Leakage Static Analysis [Drobnjaković24]

3-Step Recipe

practical tools
targeting specific programs



abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Experimental Evaluation

7378 Executions in 2111 Notebooks from Kaggle

Implementation	True Positives		False Positives
	Taint Data Leakage	Overlap Data Leakage	
NBLyzer + Original Data Leakage Analysis	10	0	2
NBLyzer + Our Data Leakage Analysis	**10**	**15**	**2**

IN 5 NOTEBOOKS

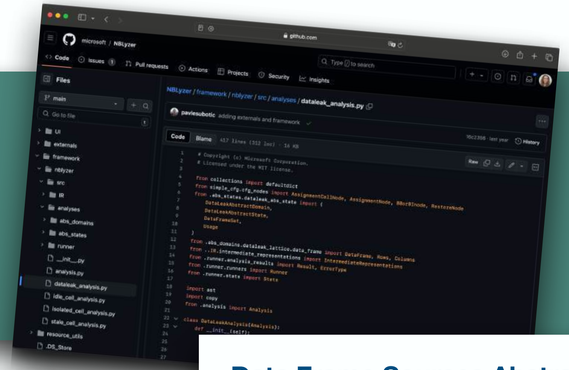
IN 11 NOTEBOOKS

CONFIRMED BY 4 DATA SCIENTISTS AT MICROSOFT

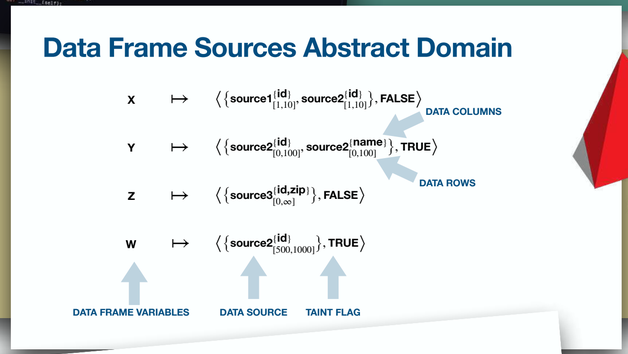
Data Leakage Static Analysis [Drobnjaković24]

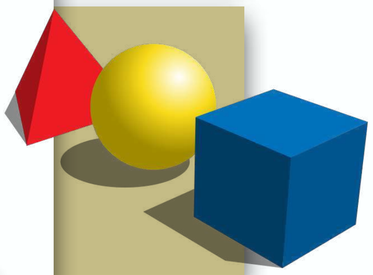
3-Step Recipe

practical tools
targeting specific programs

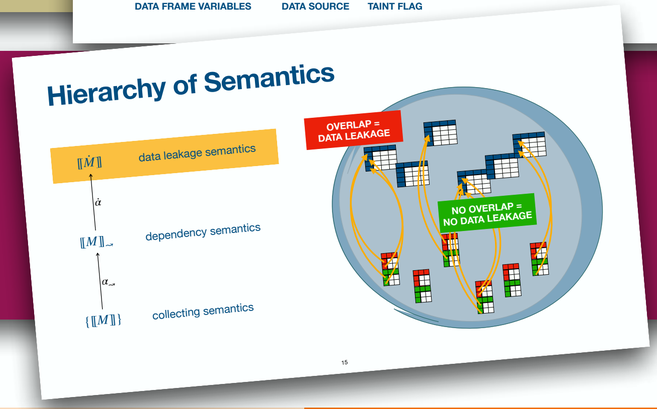


abstract semantics, abstract domains
algorithmic approaches to decide program properties



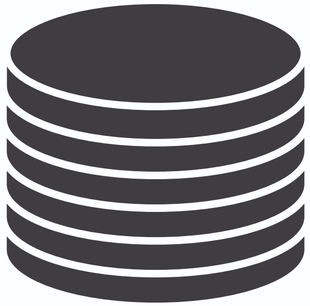


concrete semantics
mathematical models of the program behavior



Unexpected Data

Unexpected Data



missing values

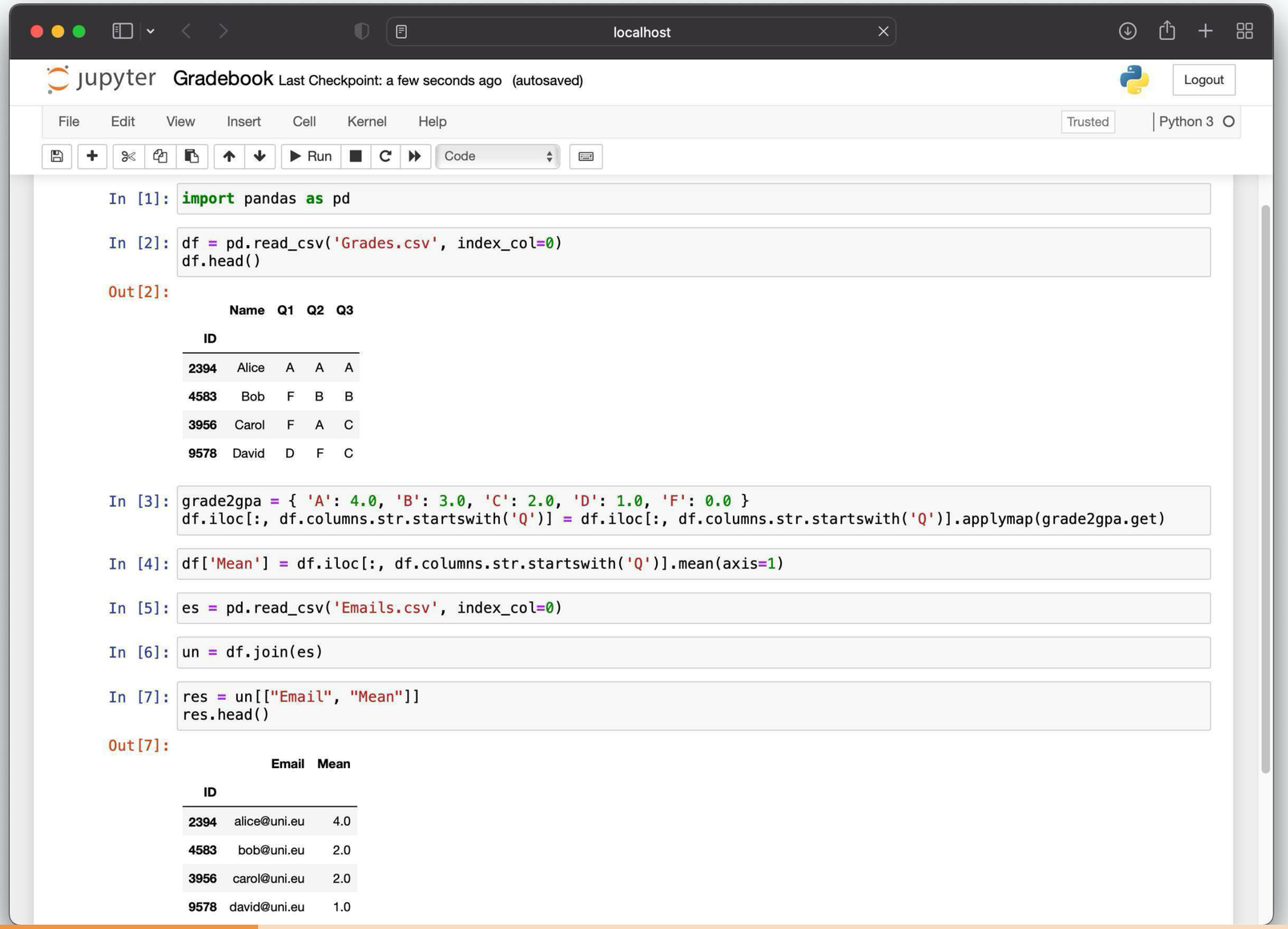
extra values

duplicate values

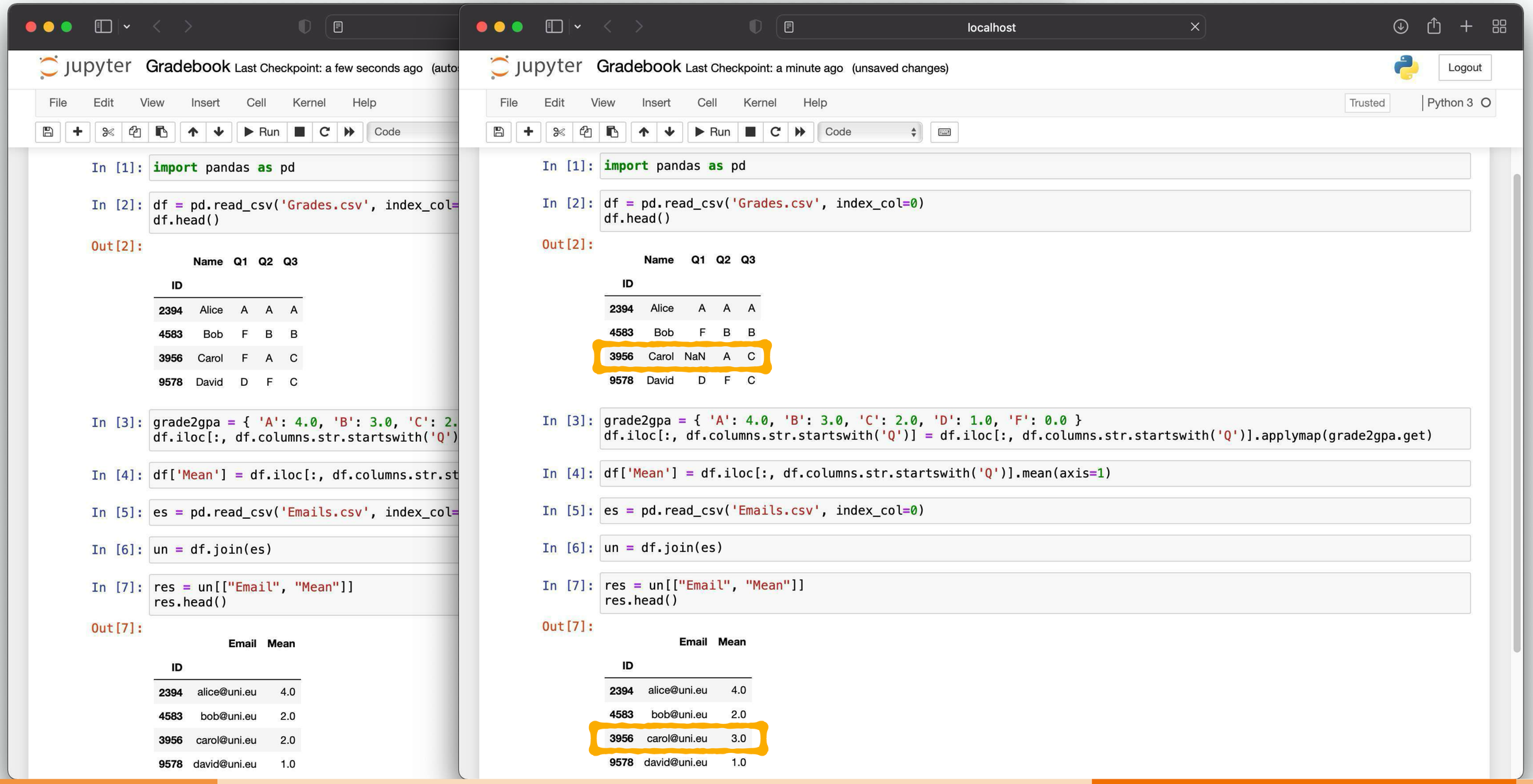
different **format**

different data

Example


localhost
jupyter Gradebook Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
In [1]:
import pandas as pd
In [2]:
df = pd.read_csv('Grades.csv', index_col=0)
df.head()
Out[2]:
Name Q1 Q2 Q3
ID
2394 Alice A A A
4583 Bob F B B
3956 Carol F A C
9578 David D F C
In [3]:
grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
df.iloc[:, df.columns.str.startswith('Q')] = df.iloc[:, df.columns.str.startswith('Q')].applymap(grade2gpa.get)
In [4]:
df['Mean'] = df.iloc[:, df.columns.str.startswith('Q')].mean(axis=1)
In [5]:
es = pd.read_csv('Emails.csv', index_col=0)
In [6]:
un = df.join(es)
In [7]:
res = un[["Email", "Mean"]]
res.head()
Out[7]:
Email Mean
ID
2394 alice@uni.eu 4.0
4583 bob@uni.eu 2.0
3956 carol@uni.eu 2.0
9578 david@uni.eu 1.0

Missing Values



jupyter Gradebook Last Checkpoint: a few seconds ago (auto

File Edit View Insert Cell Kernel Help

Run Code

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('Grades.csv', index_col=
        df.head()
```

Out[2]:

ID	Name	Q1	Q2	Q3
2394	Alice	A	A	A
4583	Bob	F	B	B
3956	Carol	F	A	C
9578	David	D	F	C

```
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.
        df.iloc[:, df.columns.str.startswith('Q')

In [4]: df['Mean'] = df.iloc[:, df.columns.str.st

In [5]: es = pd.read_csv('Emails.csv', index_col=

In [6]: un = df.join(es)

In [7]: res = un[["Email", "Mean"]]
        res.head()
```

Out[7]:

ID	Email	Mean
2394	alice@uni.eu	4.0
4583	bob@uni.eu	2.0
3956	carol@uni.eu	2.0
9578	david@uni.eu	1.0

localhost

jupyter Gradebook Last Checkpoint: a minute ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help

Trusted Python 3

Run Code

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('Grades.csv', index_col=0)
        df.head()
```

Out[2]:

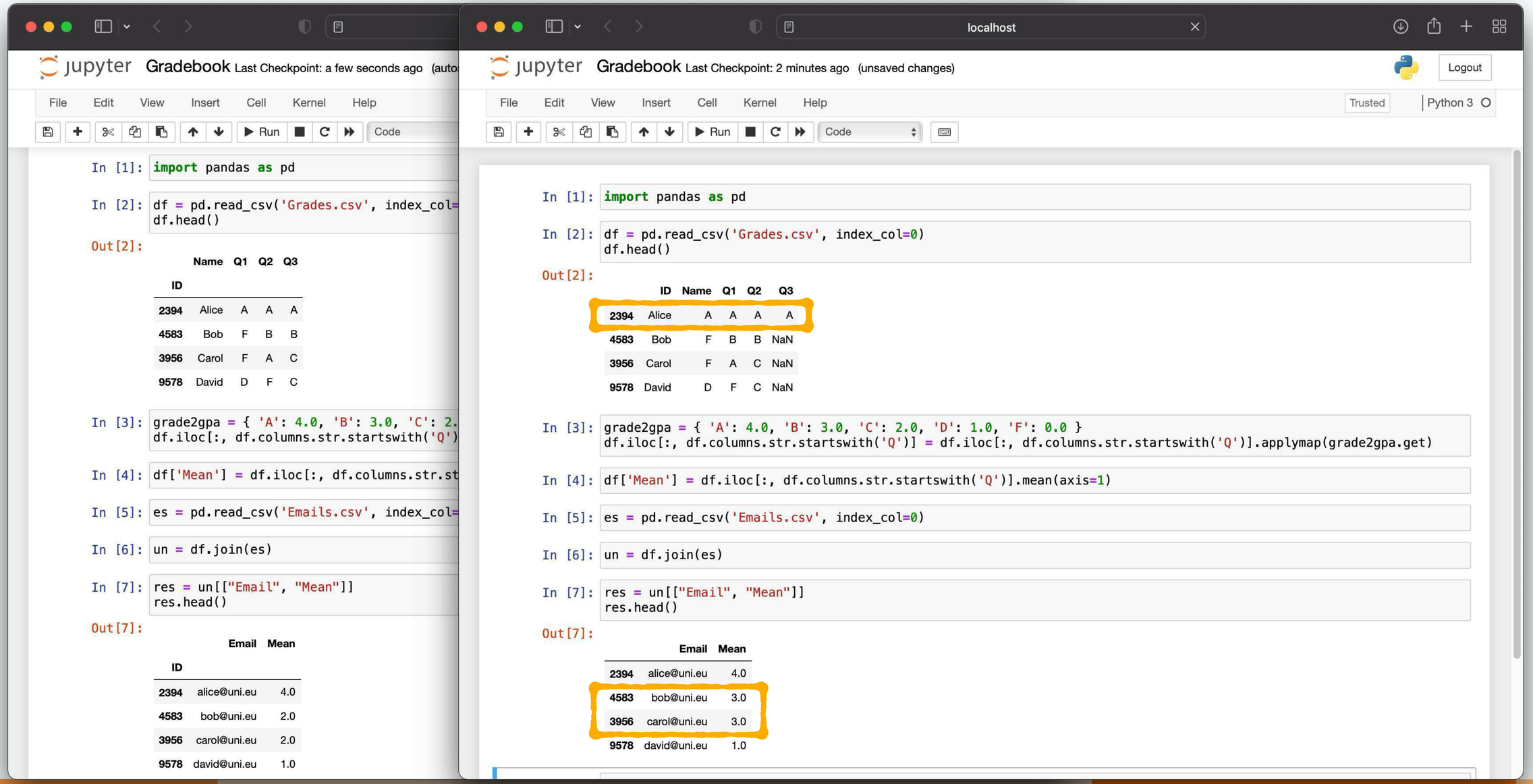
ID	Name	Q1	Q2	Q3
2394	Alice	A	A	A
4583	Bob	F	B	B
3956	Carol	NaN	A	C
9578	David	D	F	C

```
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
        df.iloc[:, df.columns.str.startswith('Q')] = df.iloc[:, df.columns.str.startswith('Q')].applymap(grade2gpa.get)

In [4]: df['Mean'] = df.iloc[:, df.columns.str.startswith('Q')].mean(axis=1)

In [5]: es = pd.read_csv('Emails.csv', index_col=0)

In [6]: un = df.join(es)

In [7]: res = un[["Email", "Mean"]]
        res.head()
```

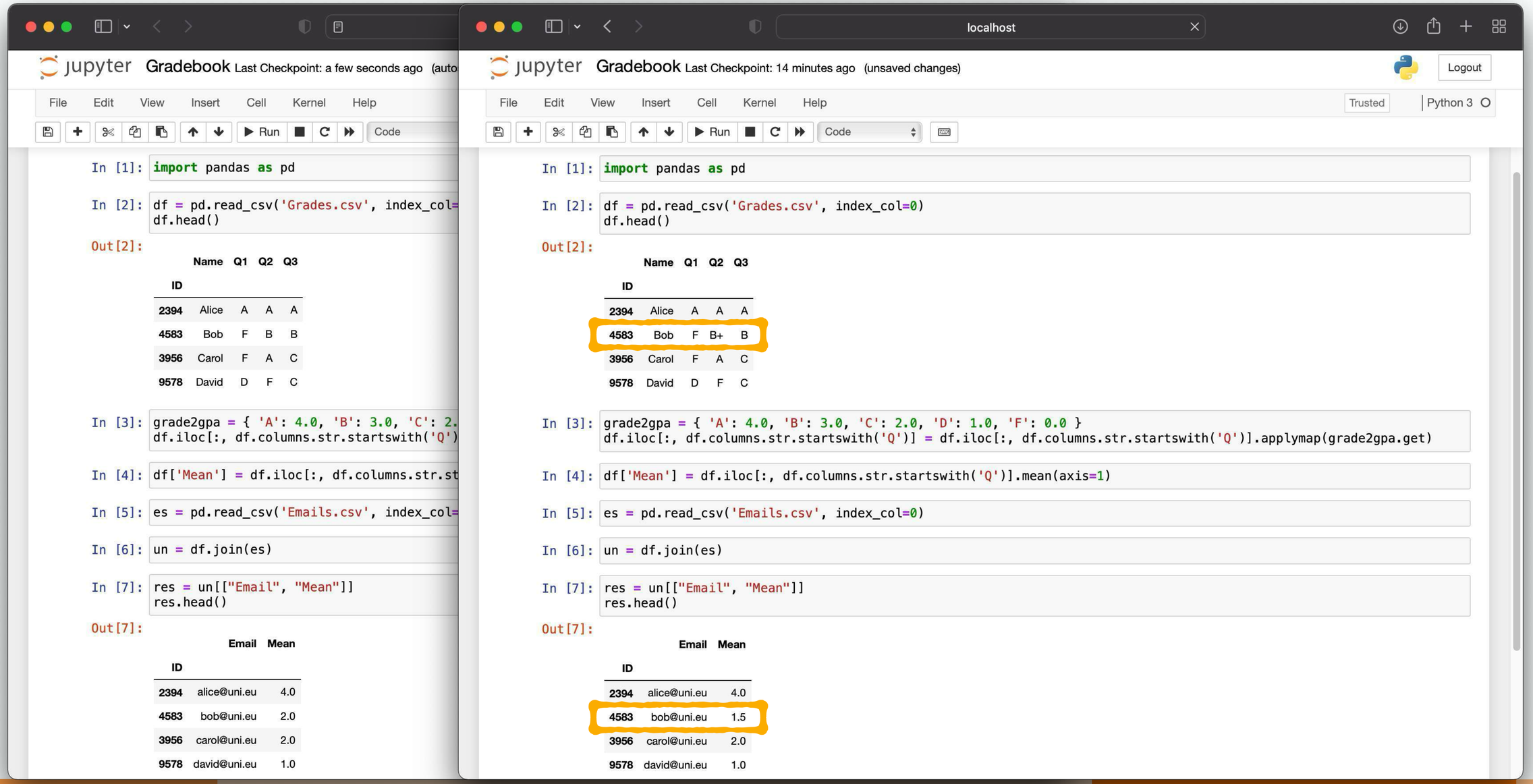
Out[7]:

ID	Email	Mean
2394	alice@uni.eu	4.0
4583	bob@uni.eu	2.0
3956	carol@uni.eu	3.0
9578	david@uni.eu	1.0

Extra Values



Different Format



jupyter Gradebook Last Checkpoint: a few seconds ago (auto

File Edit View Insert Cell Kernel Help

Run Code

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('Grades.csv', index_col=
        df.head()
```

Out[2]:

ID	Name	Q1	Q2	Q3
2394	Alice	A	A	A
4583	Bob	F	B	B
3956	Carol	F	A	C
9578	David	D	F	C

```
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.
        df.iloc[:, df.columns.str.startswith('Q')

In [4]: df['Mean'] = df.iloc[:, df.columns.str.st

In [5]: es = pd.read_csv('Emails.csv', index_col=

In [6]: un = df.join(es)

In [7]: res = un[["Email", "Mean"]]
        res.head()
```

Out[7]:

ID	Email	Mean
2394	alice@uni.eu	4.0
4583	bob@uni.eu	2.0
3956	carol@uni.eu	2.0
9578	david@uni.eu	1.0

localhost

jupyter Gradebook Last Checkpoint: 14 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help

Trusted Python 3

Run Code

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('Grades.csv', index_col=0)
        df.head()
```

Out[2]:

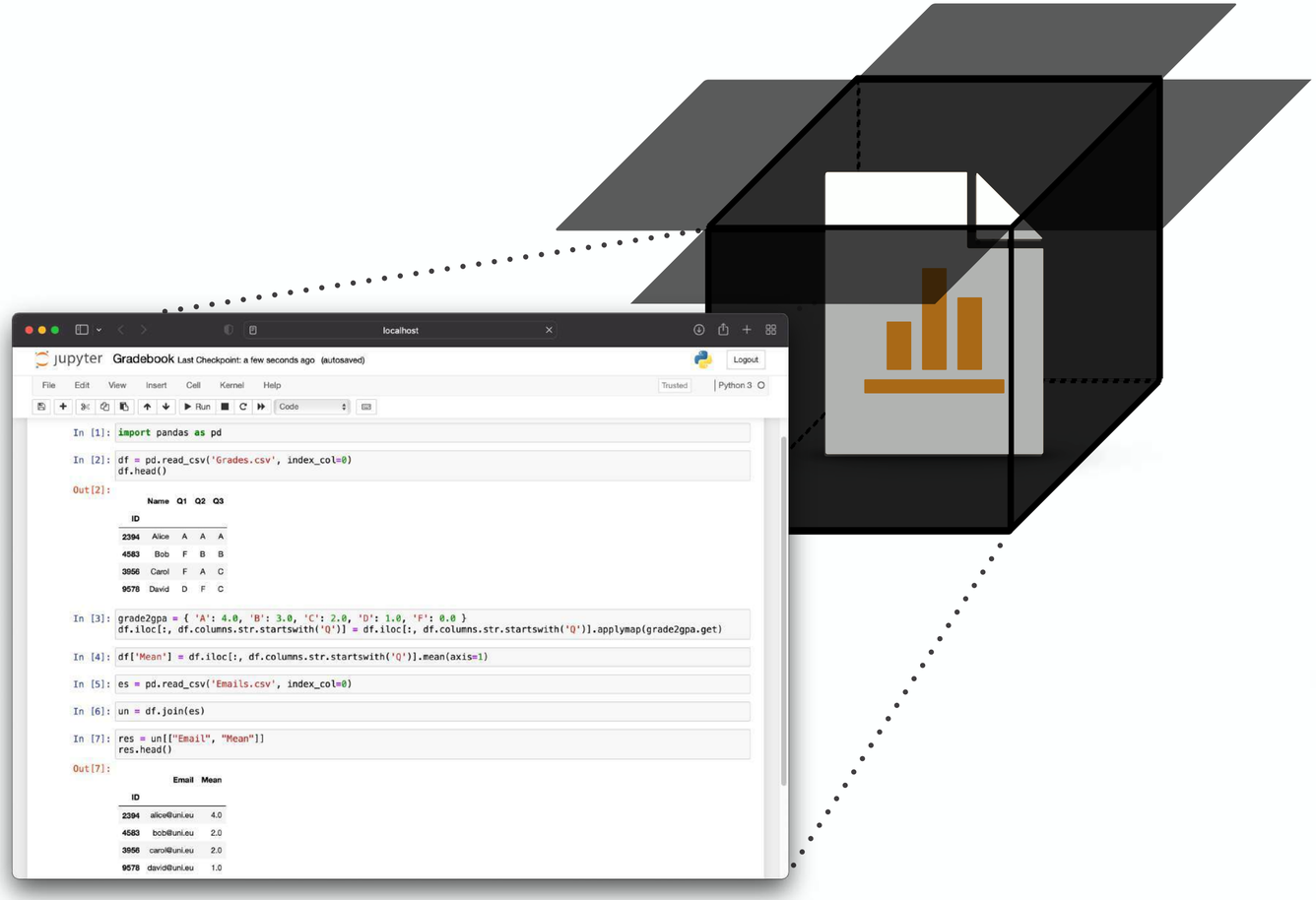
ID	Name	Q1	Q2	Q3
2394	Alice	A	A	A
4583	Bob	F	B+	B
3956	Carol	F	A	C
9578	David	D	F	C

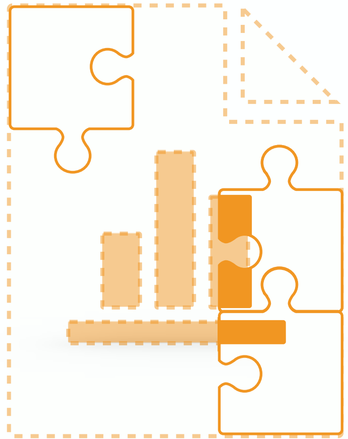
```
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
        df.iloc[:, df.columns.str.startswith('Q')] = df.iloc[:, df.columns.str.startswith('Q')].applymap(grade2gpa.get)

In [4]: df['Mean'] = df.iloc[:, df.columns.str.startswith('Q')].mean(axis=1)

In [5]: es = pd.read_csv('Emails.csv', index_col=0)

In [6]: un = df.join(es)

In [7]: res = un[["Email", "Mean"]]
        res.head()
```

Out[7]:

ID	Email	Mean
2394	alice@uni.eu	4.0
4583	bob@uni.eu	1.5
3956	carol@uni.eu	2.0
9578	david@uni.eu	1.0

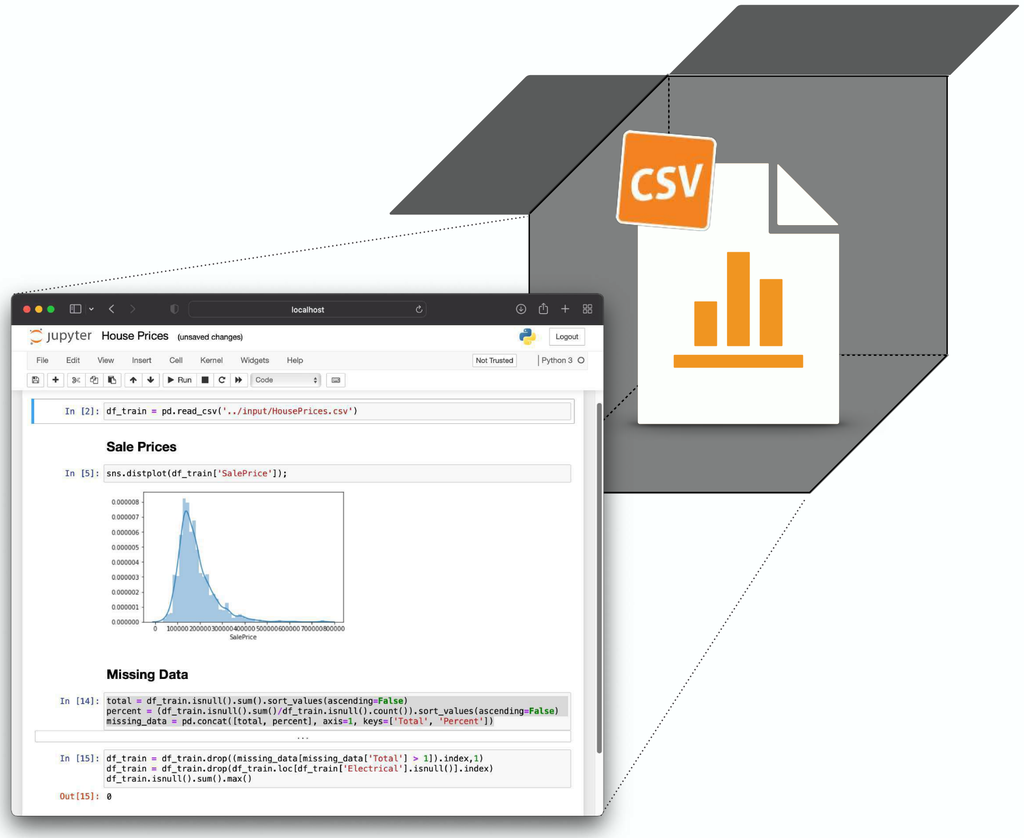
Data Expectations Static Analysis





Data

1st Challenge: Multi-Dimensional Data Structures

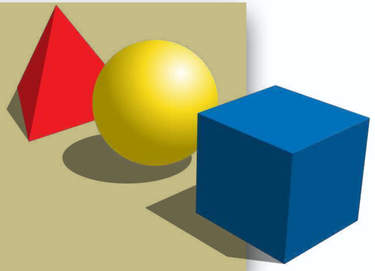


Data Expectations Static Analysis

3-Step Recipe

practical tools
targeting specific programs

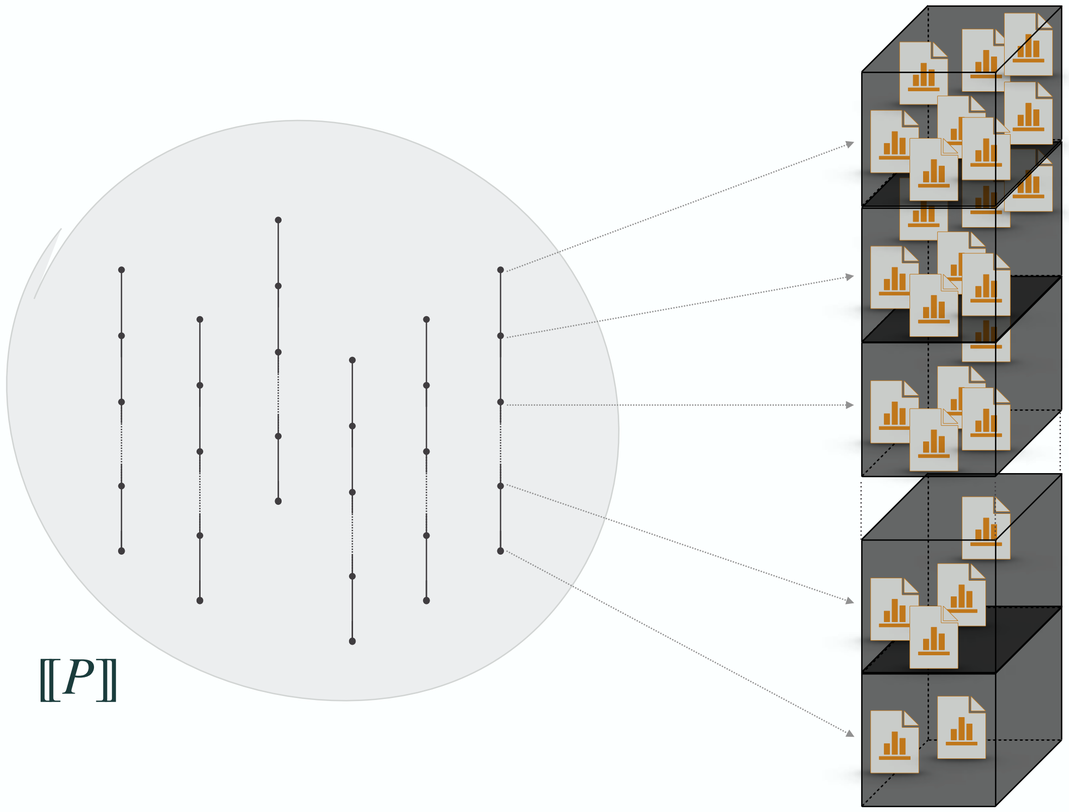
abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Concrete Semantics

2nd Challenge: Indirect Reasoning



Abstract Semantics

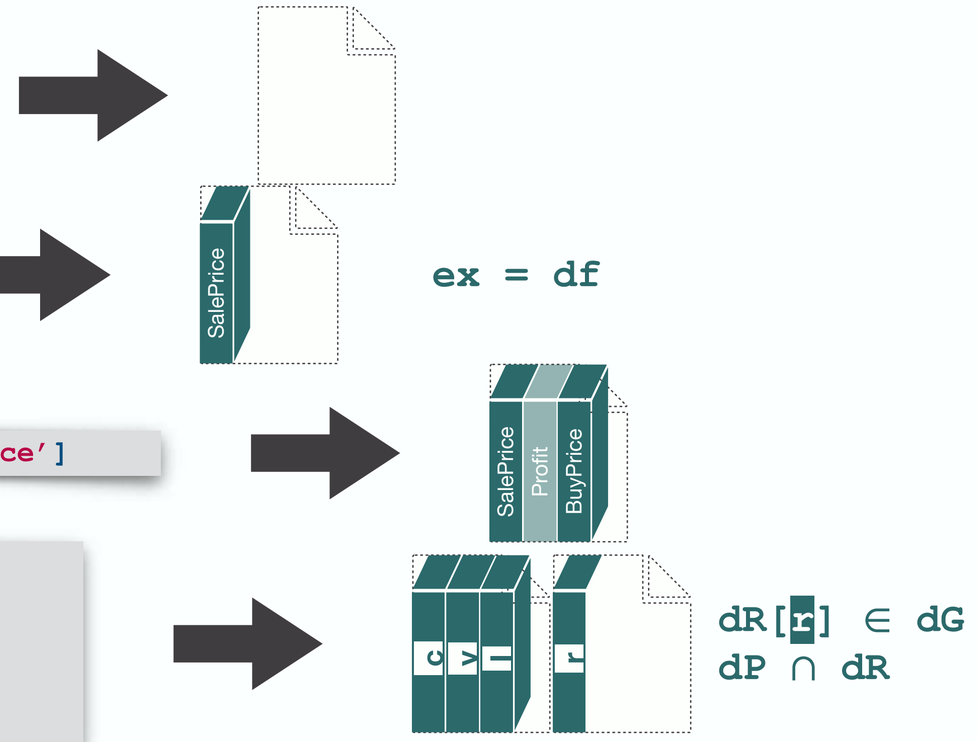
3rd Challenge: Complex Library Calls

```
import pandas as pd
df = pd.read_csv("HousePrices.csv")
```

```
ex = df[df.SalePrice >= 1000000]
```

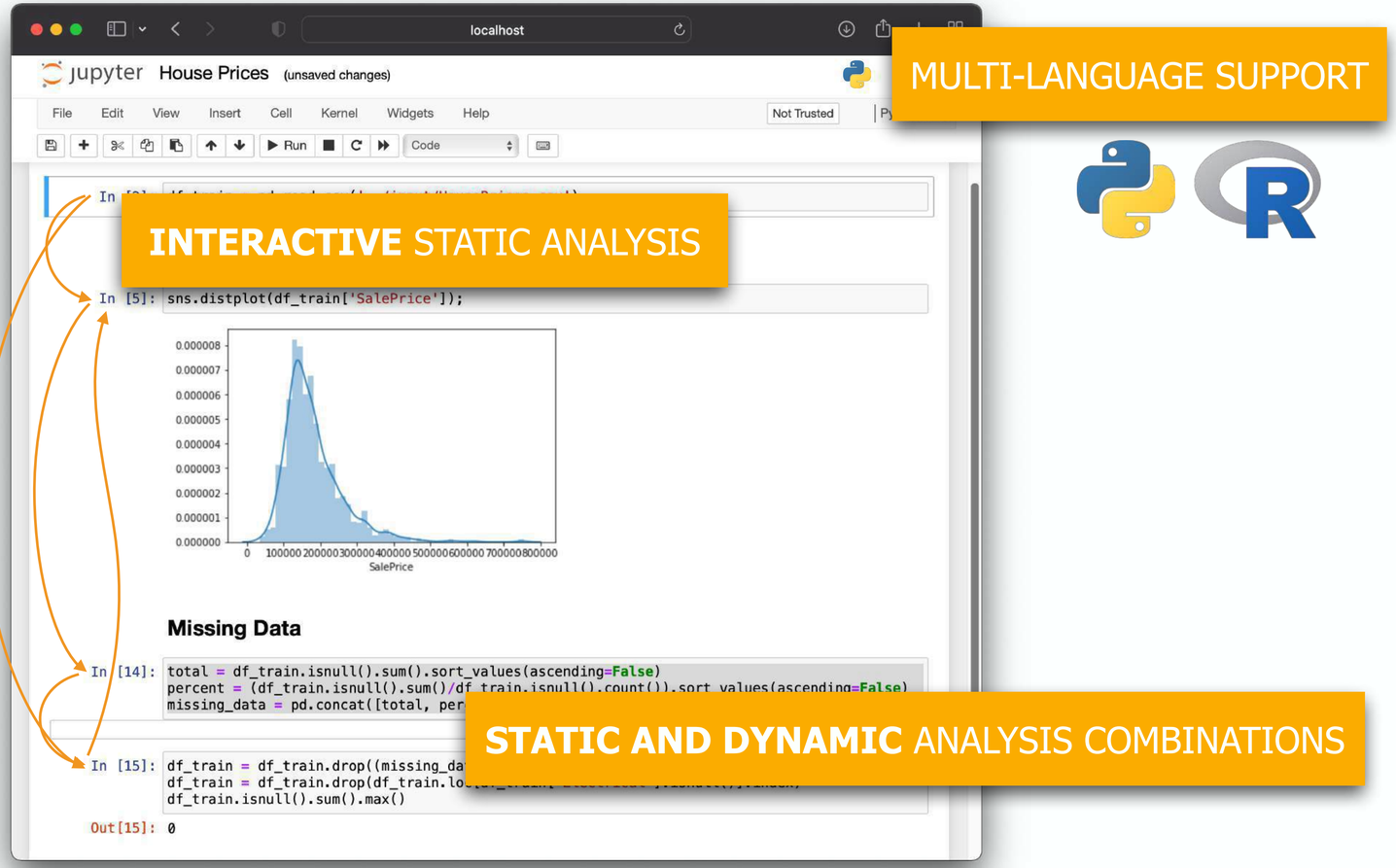
```
ex['Profit'] = ex['SalePrice'] - ex['BuyPrice']
```

```
⋮
dL = pd.read_csv("L.csv")
dP = dL.pivot(index=c, columns=y, values=l)
dR = pd.read_csv("R.csv")
dG = dP.loc[:, 0:35].groupby(dR[r])
```

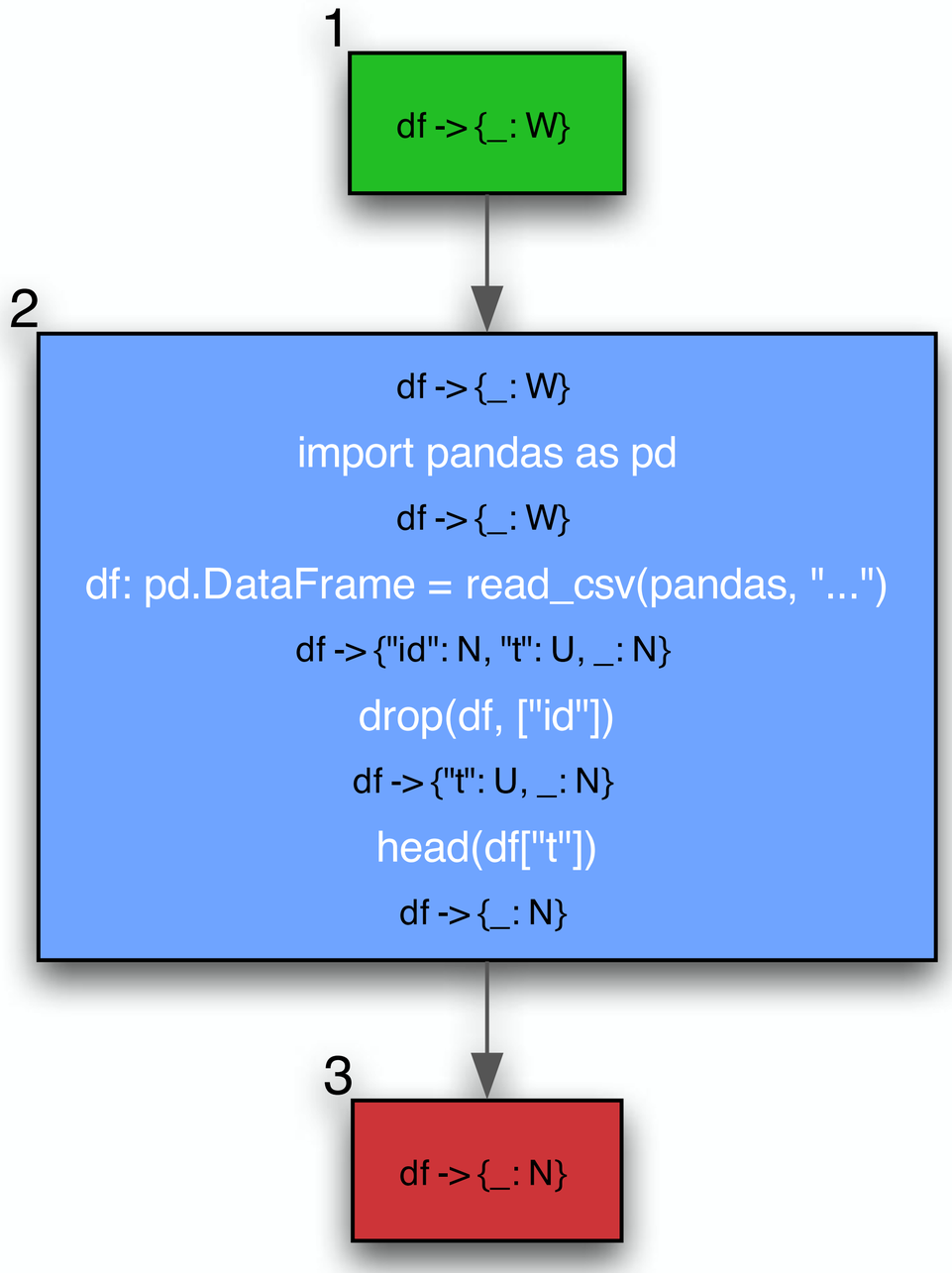


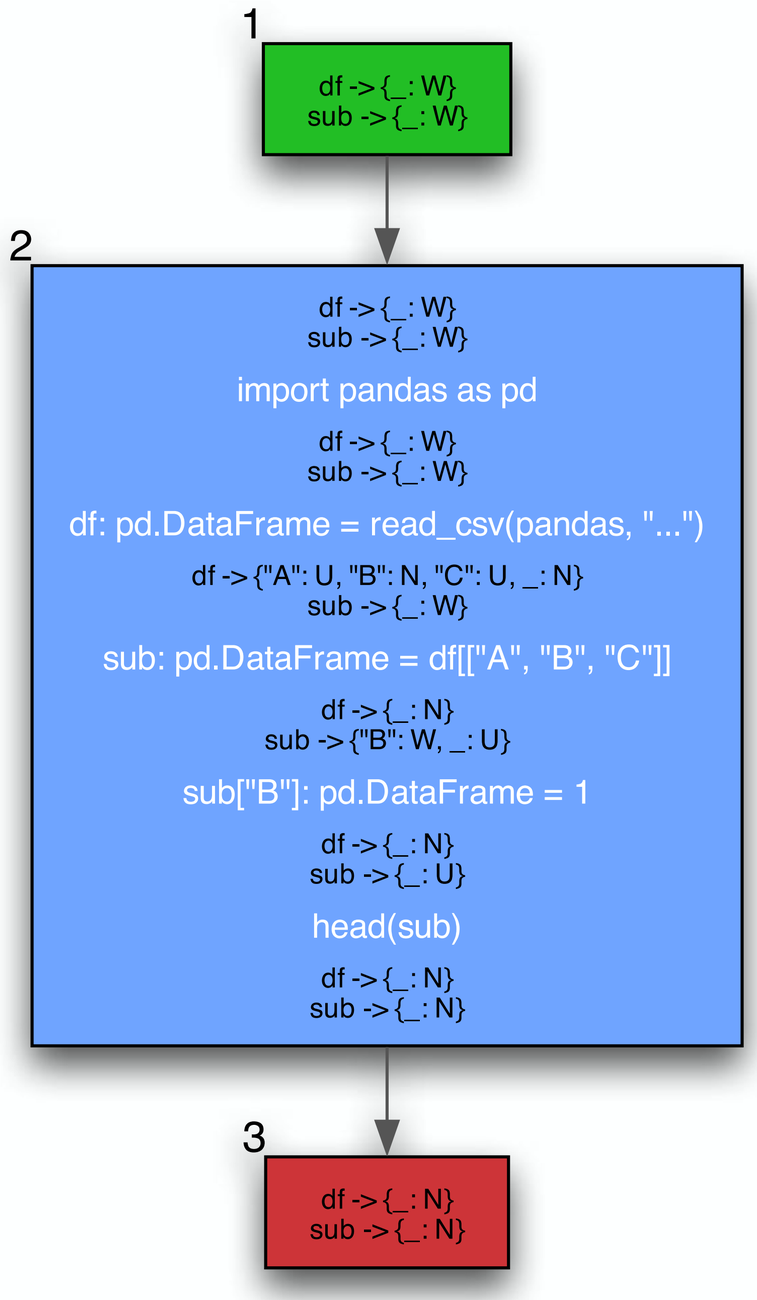
Implementation

Wish List



(Un)expected + (Un)used Data


1
df -> {_: W}
2
df -> {_: W}
import pandas as pd
df -> {_: W}
df: pd.DataFrame = read_csv(pandas, "...")
df -> {"id": N, "t": U, _: N}
drop(df, ["id"])
df -> {"t": U, _: N}
head(df["t"])
df -> {_: N}
3
df -> {_: N}


1
df -> {_: W}
sub -> {_: W}
2
df -> {_: W}
sub -> {_: W}
import pandas as pd
df -> {_: W}
sub -> {_: W}
df: pd.DataFrame = read_csv(pandas, "...")
df -> {"A": U, "B": N, "C": U, _: N}
sub -> {_: W}
sub: pd.DataFrame = df[["A", "B", "C"]]
df -> {_: N}
sub -> {"B": W, _: U}
sub["B"]: pd.DataFrame = 1
df -> {_: N}
sub -> {_: U}
head(sub)
df -> {_: N}
sub -> {_: N}
3
df -> {_: N}
sub -> {_: N}

Bibliography

[Urban18] **Caterina Urban and Peter Müller**. An Abstract Interpretation Framework for Data Usage. In ESOP, pages 683-710, 2018.
data (non-) usage

[Drobnjaković24] **Filip Drobnjaković, Pavle Subotić, and Caterina Urban**. An Abstract Interpretation-Based Data Leakage Static Analysis. In TASE, pages 109-126, 2024.
data leakage static analysis