An Abstract Interpretation Framework for Input Data Usage

Caterina Urban and Peter Müller

Department of Computer Science
ETH Zürich, Switzerland

Data is Revolutionizing Industries

Data Science Pipeline

data sources

stored data

pre-processing

data analysis

results

Data Science Pipeline

data sources → **stored data** → **pre**

mislabeled data

accidentally duplicated data

wrongly converted data

accidentally unused data

The New York Times | https://nyti.ms/1mZywng

TECHNOLOGY

For Big-Data Scientists, ‘Janitor Work’ Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014

Technology revolutions come in measured, sometimes foot-dragging steps. The lab science and marketing enthusiasm tend to underestimate the bottlenecks to progress that must be overcome with hard work and practical engineering.

The field known as “big data” offers a contemporary case study. The catchphrase stands for the modern abundance of digital data from many sources — the web, sensors, smartphones and corporate databases — that can be mined with clever software for discoveries and insights. Its promise is smarter, data-driven decision-making in every field. That is why data scientist is the economy’s hot new job.

Yet far too much handcrafted work — what data scientists call “data wrangling,” “data munging” and “data janitor work” — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

most errors are made in the pre-processing phase

The Reinhart-Rogoff Paper

American Economic Review: Papers & Proceedings 100 (May 2010): 573–578
http://www.aeaweb.org/articles.php?doi=10.1257/aer.100.2.573

Growth in a Time of Debt

By Carmen M. Reinhart and Kenneth S. Rogoff*

In this paper, we exploit a new multi-country historical dataset on public (government) debt to search for a systemic relationship between high public debt levels, growth and inflation.[1] Our main result is that whereas the link between growth and debt seems relatively weak at "normal" debt levels, median growth rates for countries with public debt over roughly 90 percent of GDP are about one percent lower than otherwise; average (mean) growth rates are several percent lower. Surprisingly, the relationship between public debt and growth is remarkably similar across emerging markets and advanced economies. This is not the case for inflation. We find no systematic relationship between high debt levels and inflation for advanced economies as a group (albeit with individual country exceptions including the United States). By contrast, in emerging market countries, high public debt levels coincide with higher inflation.

Our topic would seem to be a timely one. Public debt has been soaring in the wake of the recent global financial maelstrom, especially in the epicenter countries. This should not be surprising, given the experience of earlier severe financial crises.[2] Outsized deficits and epic bank bailouts may be useful in fighting a downturn, but what is the long-run macroeconomic impact, especially against the backdrop of graying populations and rising social insurance costs? Are sharply elevated public debts ultimately a manageable policy challenge?

Our approach here is decidedly empirical, taking advantage of a broad new historical dataset on public debt (in particular, central government debt) first presented in Carmen M. Reinhart and Kenneth S. Rogoff (2008, 2009b). Prior to this dataset, it was exceedingly difficult to get more than two or three decades of public debt data even for many rich countries, and virtually impossible for most emerging markets. Our results incorporate data on 44 countries spanning about 200 years. Taken together, the data incorporate over 3,700 annual observations covering a wide range of political systems, institutions, exchange rate and monetary arrangements, and historic circumstances.

We also employ more recent data on external debt, including debt owed both by governments and by private entities. For emerging markets, we find that there exists a significantly more severe threshold for total gross external debt (public and private)—which is almost exclusively denominated in a foreign currency—than for total public debt (the domestically issued component of which is largely denominated in home currency). When gross external debt reaches 60 percent of GDP, annual growth declines by about two percent; for levels of external debt in excess of 90 percent of GDP, growth rates are roughly cut in half. We are not in a position to calculate separate total external debt thresholds (as opposed to public debt thresholds) for advanced countries. The available time-series is too recent, beginning only in 2000. We do note, however, that external debt levels in advanced countries now average nearly 200 percent of GDP, with external debt levels being particularly high across Europe.

The focus of this paper is on the longer term macroeconomic implications of much higher public and external debt. The final section, how-

*Reinhart: Department of Economics, 4115 Tydings Hall, University of Maryland, College Park, MD 20742 (e-mail: creinhar@umd.edu); Rogoff: Economics Department, 216 Littauer Center, Harvard University, Cambridge MA 02138–3001 (e-mail: krogoff@harvard.edu). The authors would like to thank Olivier Jeanne and Vincent R. Reinhart for helpful comments.

[1] In this paper "public debt" refers to gross central government debt. "Domestic public debt" is government debt issued under domestic legal jurisdiction. Public debt does not include debts carrying a government guarantee. Total gross external debt includes the external debts of *all* branches of government as well as private debt that is issued by domestic private entities under a foreign jurisdiction.

[2] Reinhart and Rogoff (2009a, b) demonstrate that the aftermath of a deep financial crisis typically involves a protracted period of macroeconomic adjustment, particularly in employment and housing prices. On average, public

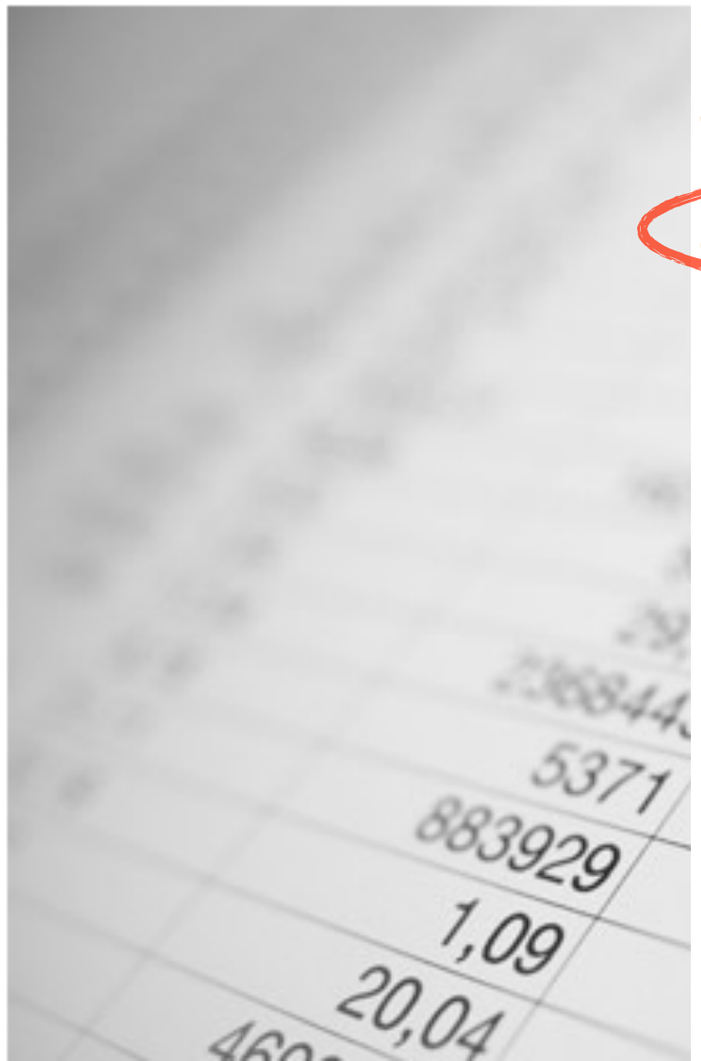
	B	C	I	J	K	L	M
2			Real GDP growth				
3			Debt/GDP				
4	Country	Coverage	30 or less	30 to 60	60 to 90	90 or above	30 or less
26			3.7	3.0	3.5	1.7	5.5
27	Minimum		1.6	0.3	1.3	-1.8	0.8
28	Maximum		5.4	4.9	10.2	3.6	13.3
29							
30	US	1946-2009	n.a.	3.4	3.3	-2.0	n.a.
31	UK	1946-2009	n.a.	2.4	2.5	2.4	n.a.
32	Sweden	1946-2009	3.6	2.9	2.7	n.a.	6.3
33	Spain	1946-2009	1.5	3.4	4.2	n.a.	9.9
34	Portugal	1952-2009	4.8	2.5	0.3	n.a.	7.9
35	New Zealand	1948-2009	2.5	2.9	3.9	-7.9	2.6
36	Netherlands	1956-2009	4.1	2.7	1.1	n.a.	6.4
37	Norway	1947-2009	3.4	5.1	n.a.	n.a.	5.4
38	Japan	1946-2009	7.0	4.0	1.0	0.7	7.0
39	Italy	1951-2009	5.4	2.1	1.8	1.0	5.6
40	Ireland	1948-2009	4.4	4.5	4.0	2.4	2.9
41	Greece	1970-2009	4.0	0.3	2.7	2.9	13.3
42	Germany	1946-2009	3.9	0.9	n.a.	n.a.	3.2
43	France	1949-2009	4.9	2.7	3.0	n.a.	5.2
44	Finland	1946-2009	3.8	2.4	5.5	n.a.	7.0
45	Denmark	1950-2009	3.5	1.7	2.4	n.a.	5.6
46	Canada	1951-2009	1.9	3.6	4.1	n.a.	2.2
47	Belgium	1947-2009	n.a.			2.6	n.a.
48	Austria	1948-2009	5.2	3.3	-3.8	n.a.	5.7
49	Australia	1951-2009	3.2	4.9	4.0	n.a.	5.9
50							
51			4.1	2.8	2.8	=AVERAGE(L30:L44)	

The Reinhart-Rogoff Paper

	B	C	I	J	K	L	M
2			Real GDP growth				
3			Debt/GDP				
4	Country	Coverage	30 or less	30 to 60	60 to 90	90 or above	30 or less
26			3.7	3.0	3.5	1.7	5.5
27	Minimum		1.6	0.3	1.3	-1.8	0.8
						3.6	13.3
						-2.0	n.a.
						2.4	n.a.
						n.a.	6.3
						n.a.	9.9
						n.a.	7.9

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?

FACEBOOK
TWITTER
GOOGLE+
SAVE
EMAIL
SHARE
PRINT
REPRINTS

Enlarge This Image

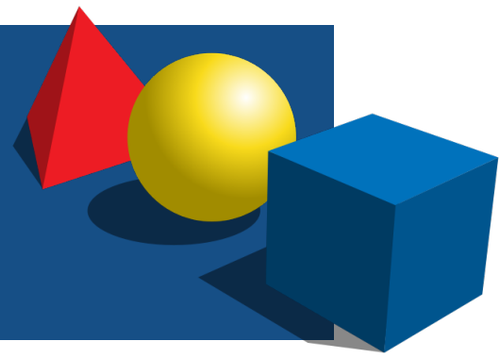
The story so far: At the beginning of 2010, two Harvard economists, Carmen Reinhart and Kenneth Rogoff, circulated a paper, "Growth in a Time of Debt," that purported to identify a critical "threshold," a tipping point, for government indebtedness. Once debt exceeds 90 percent of gross domestic product,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

Ms. Reinhart and Mr. Rogoff had credibility thanks to a widely admired earlier book on the history of financial

Static Analysis Recipe

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

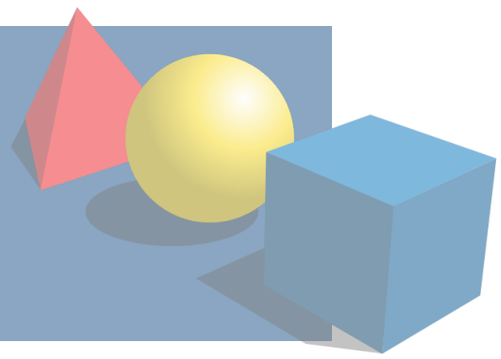


mathematical models
of the program behavior

Static Analysis Recipe

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

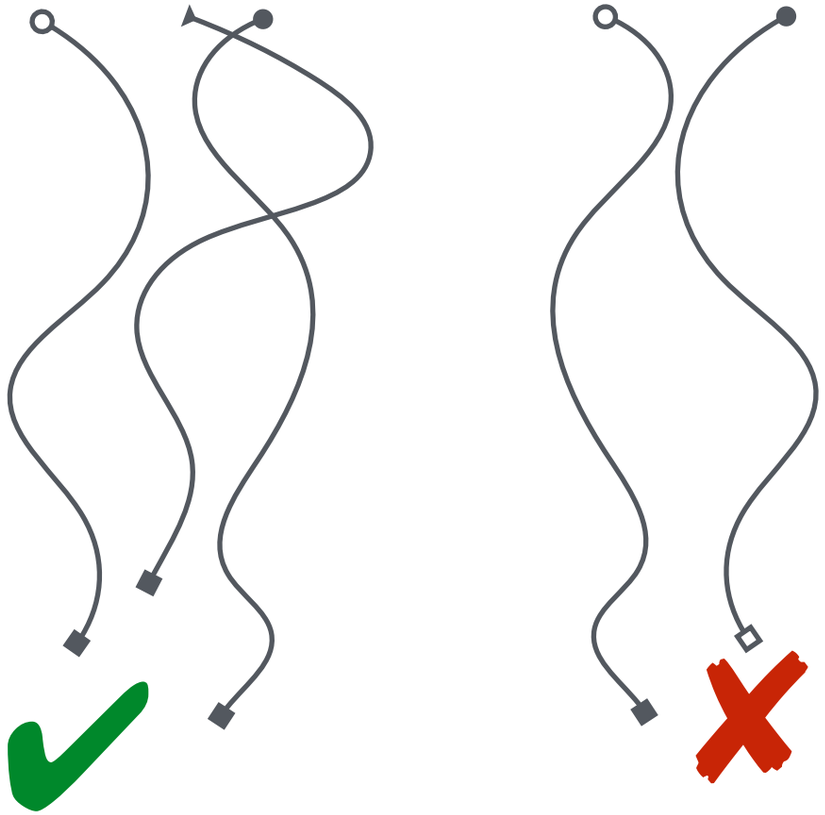


mathematical models
of the program behavior

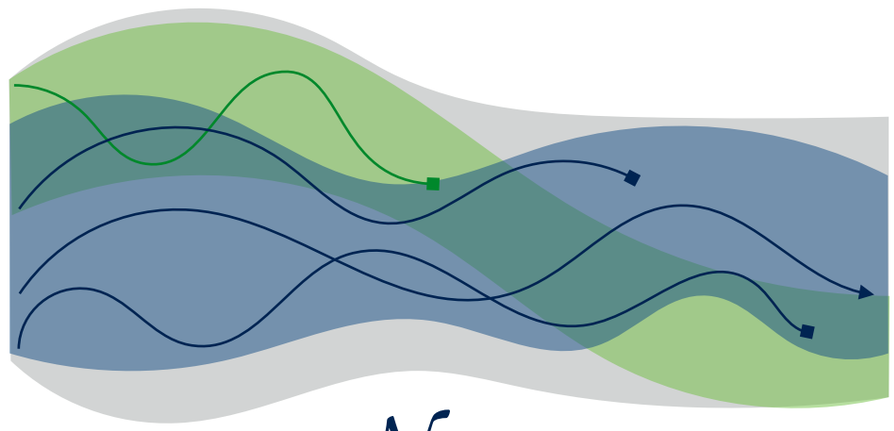
Input Data Usage is a Hyperproperty

Unused Input Data

the **outcome** of the program **does not depend** on it

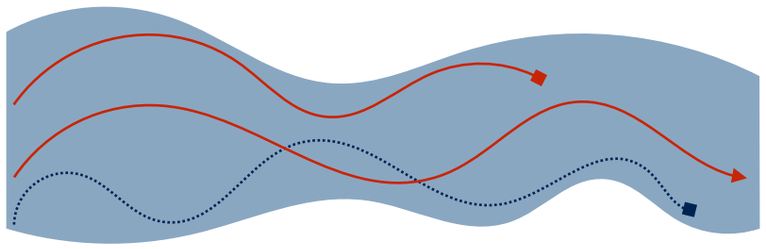


$$P \models \mathcal{N}_{\mathrm{K}} \Leftrightarrow [\![P]\!] \in \mathcal{N}_{\mathrm{K}}$$

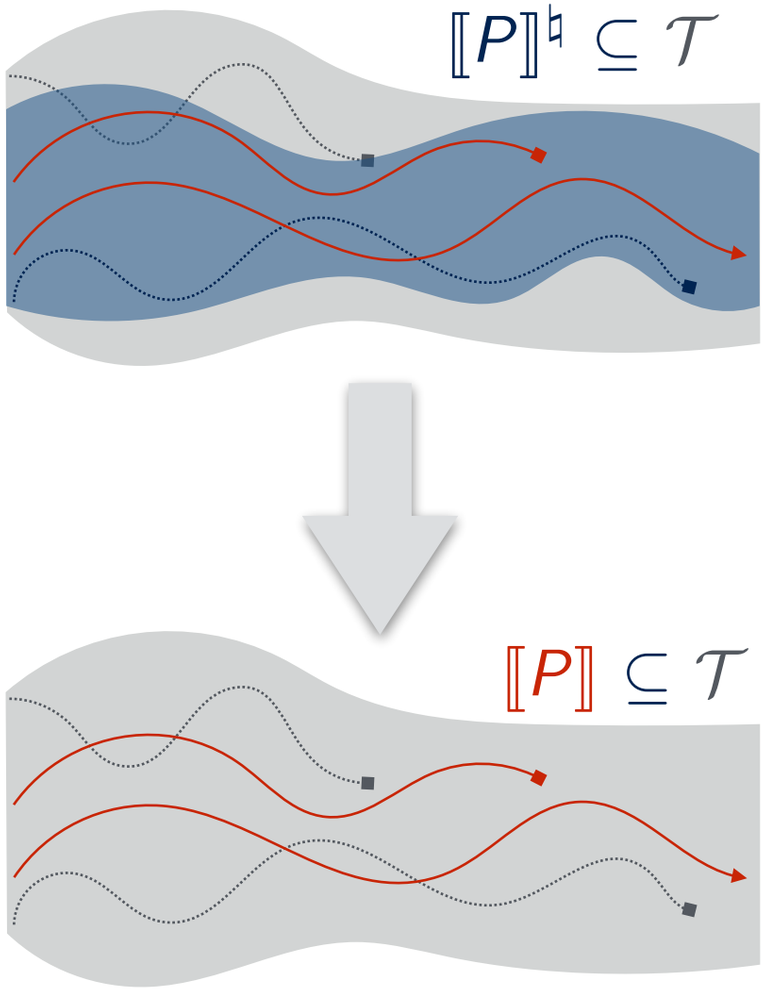


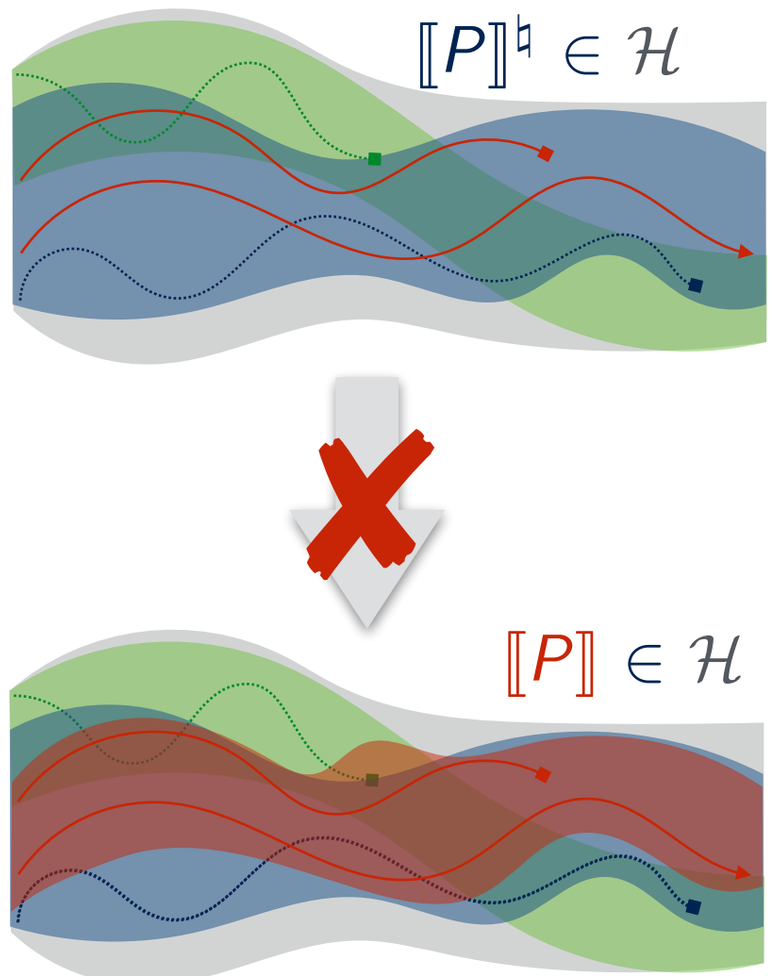
$\mathcal{N}_K$

Sound Input Data Usage Validation



$[\![P]\!] \subseteq [\![P]\!]^{\natural}$

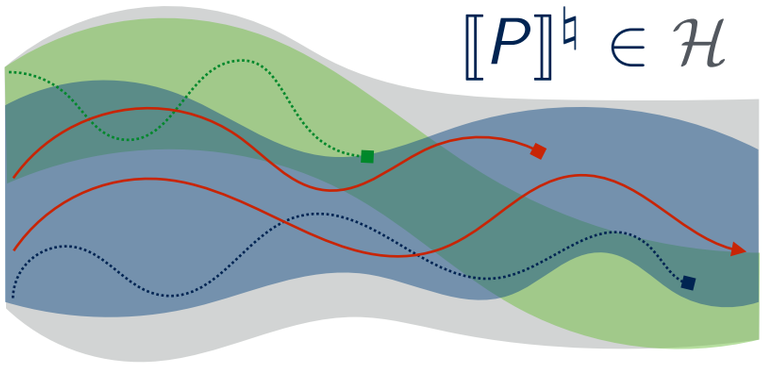


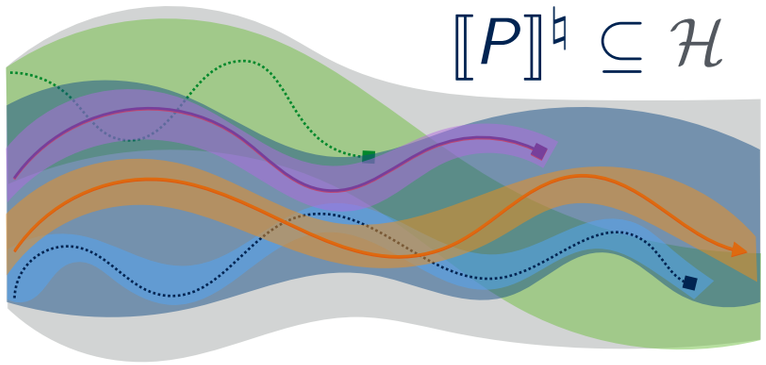


Trace Properties

Hyperproperties

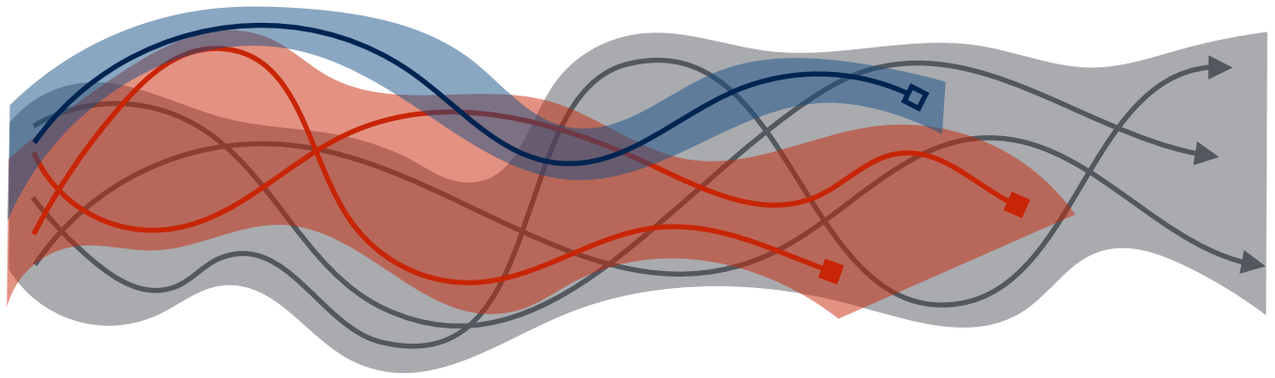
Outcome Semantics



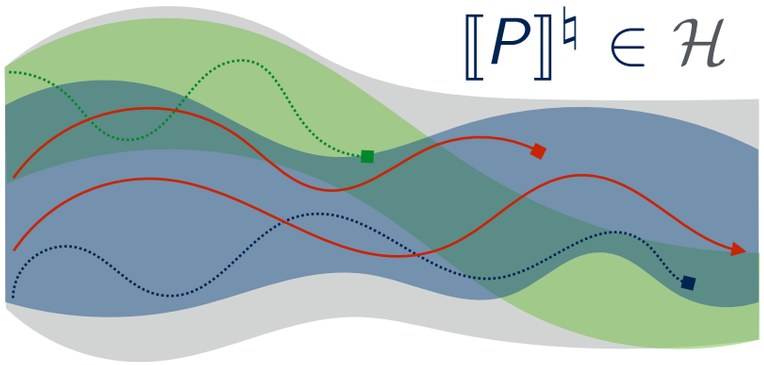


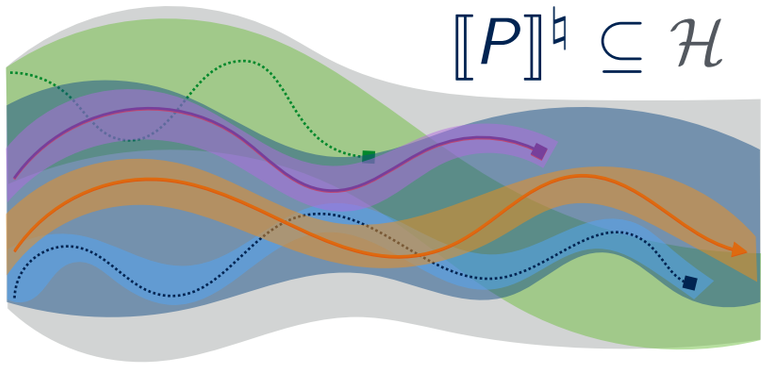
Input Data Usage

partition executions based on their outcome



Outcome Semantics

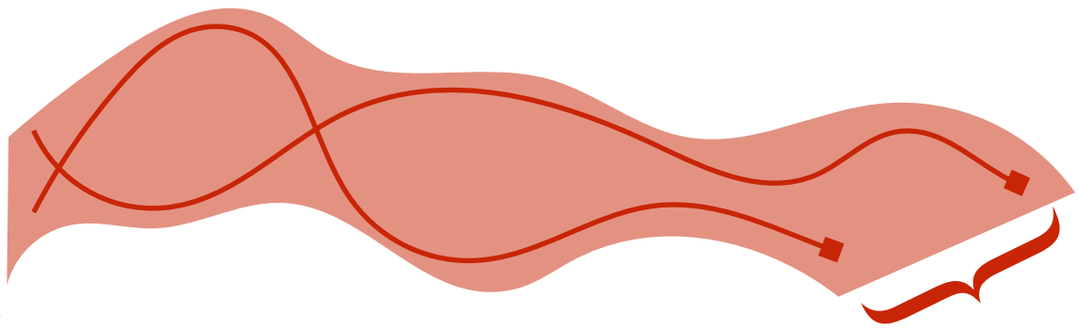




partition executions based on their outcome

Input Data Usage

any possible value of the unused data



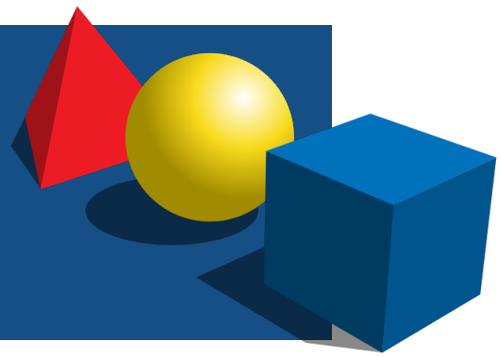
same outcome

$$P \models \mathcal{N}_{\mathrm{K}} \Leftrightarrow [\![P]\!]_{\bullet} \subseteq \mathcal{N}_{\mathrm{K}}$$

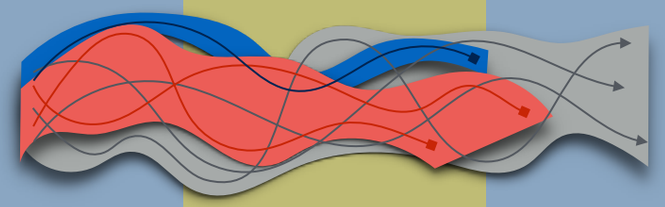
Unused Input Data Analysis

practical tools
targeting specific programs

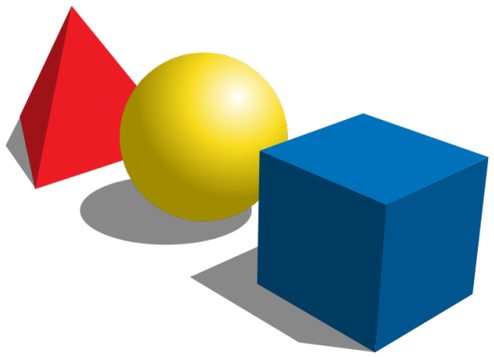
algorithmic approaches
to decide program properties



mathematical models
of the program behavior



Piecewise Unused Input Data Analysis



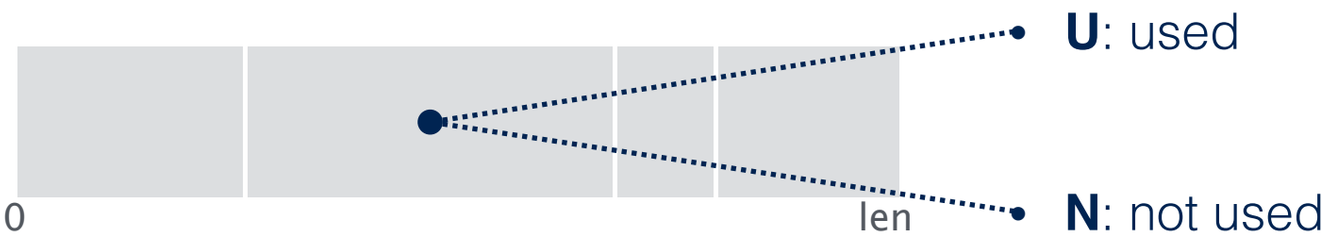
```
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):

    if grades[i] < 4:
        count = count + 1

    i = i + 1


if 2 * count < len(grades):
    passing = True
else:
    passing = False


print(passing)
```

INPUT VARIABLE

ERROR: 1 SHOULD BE 0

OUTPUT VARIABLE

Piecewise Unused Input Data Analysis



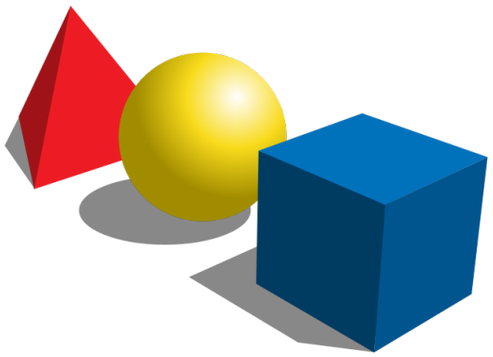

```
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):

    if grades[i] < 4:
        count = count + 1

    i = i + 1


if 2 * count < len(grades):
    passing = True
else:
    passing = False


print(passing)
```

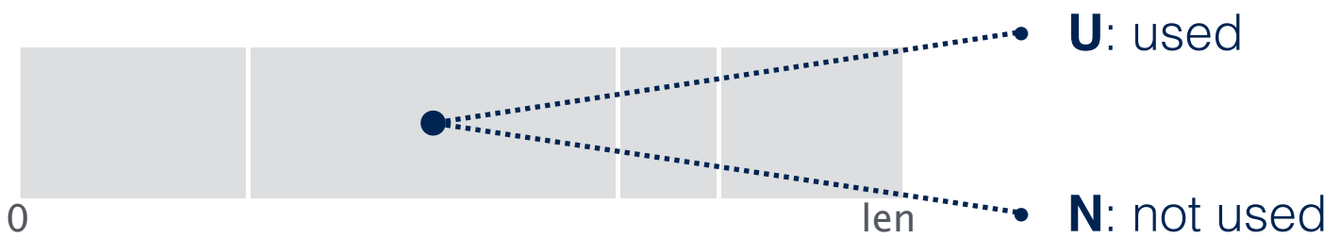
INPUT VARIABLE

ERROR: 1 SHOULD BE 0

OUTPUT VARIABLE

Piecewise Unused Input Data Analysis





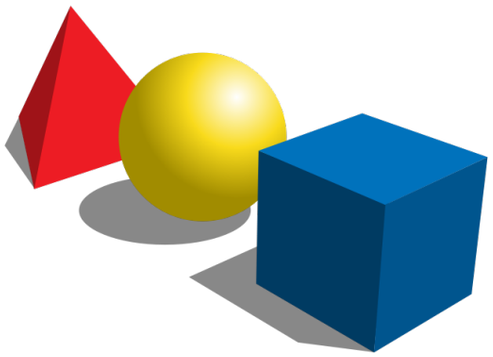

```
grades = list(map(int, input().split()))

count = 0

i = 1


while i < len(grades):

    if grades[i] < 4:
        count = count + 1

    i = i + 1


if 2 * count < len(grades):
    passing = True
else:
    passing = False


print(passing)
```

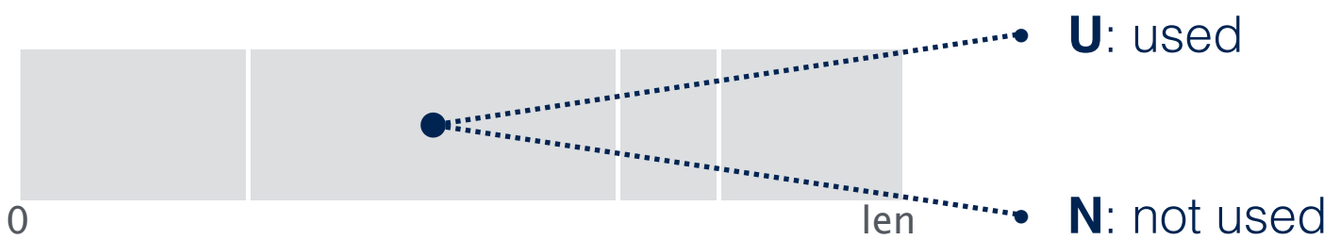
INPUT VARIABLE

ERROR: 1 SHOULD BE 0

OUTPUT VARIABLE

Piecewise Unused Input Data Analysis

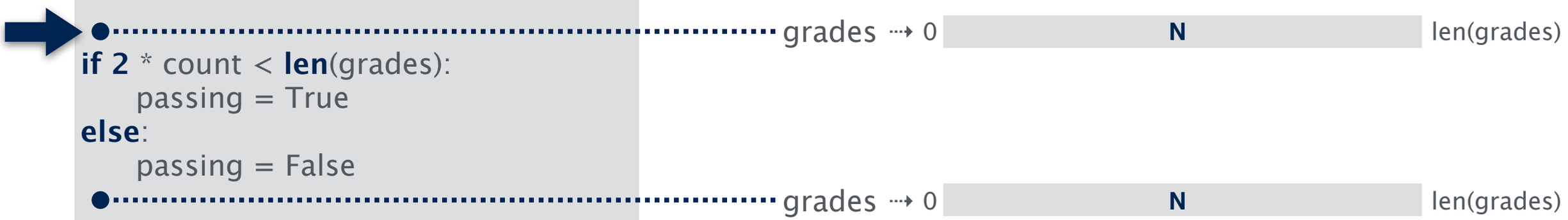





```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

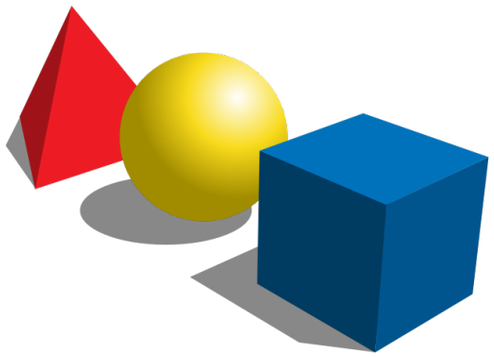
INPUT VARIABLE

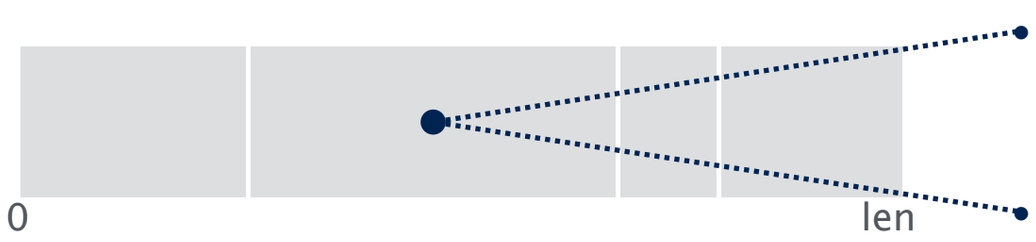
ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

Piecewise Unused Input Data Analysis





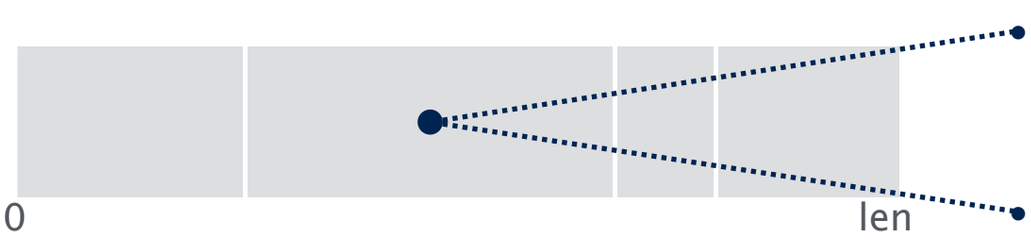

```
grades = list(map(int, input().split()))
count = 0
i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1

    i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
```

INPUT VARIABLE

ERROR: 1 SHOULD BE 0

OUTPUT VARIABLE

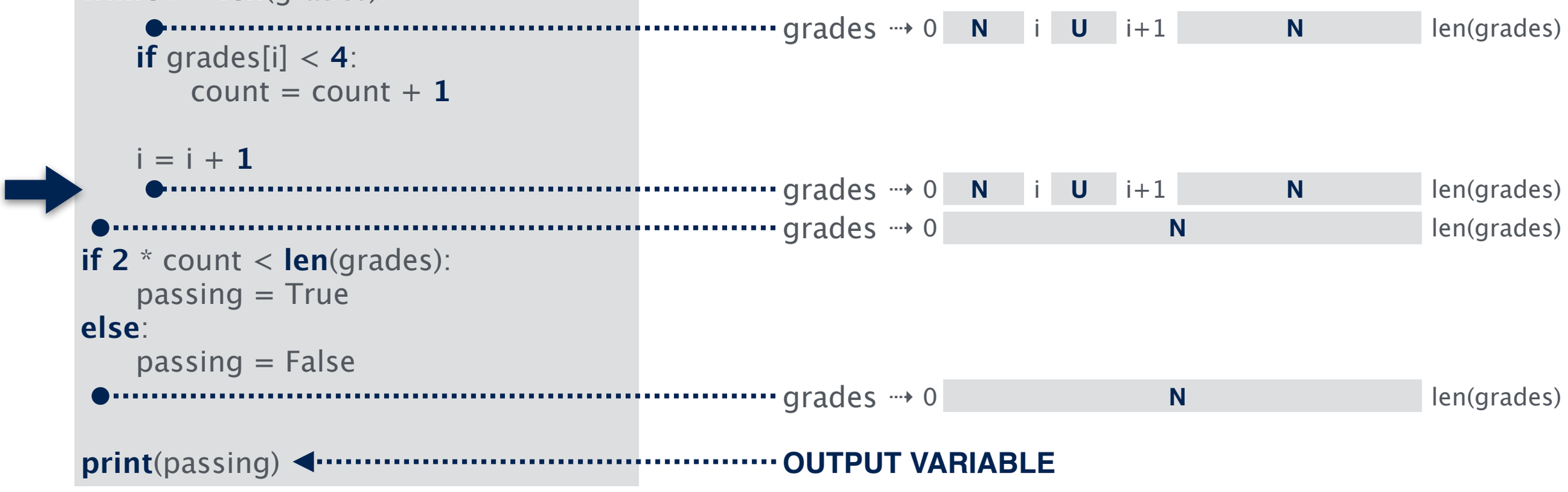
Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

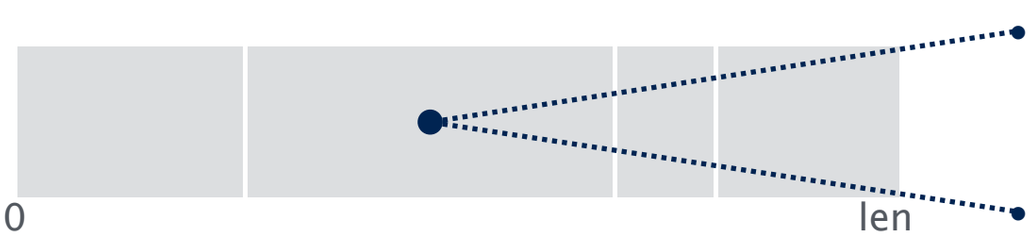
INPUT VARIABLE

ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

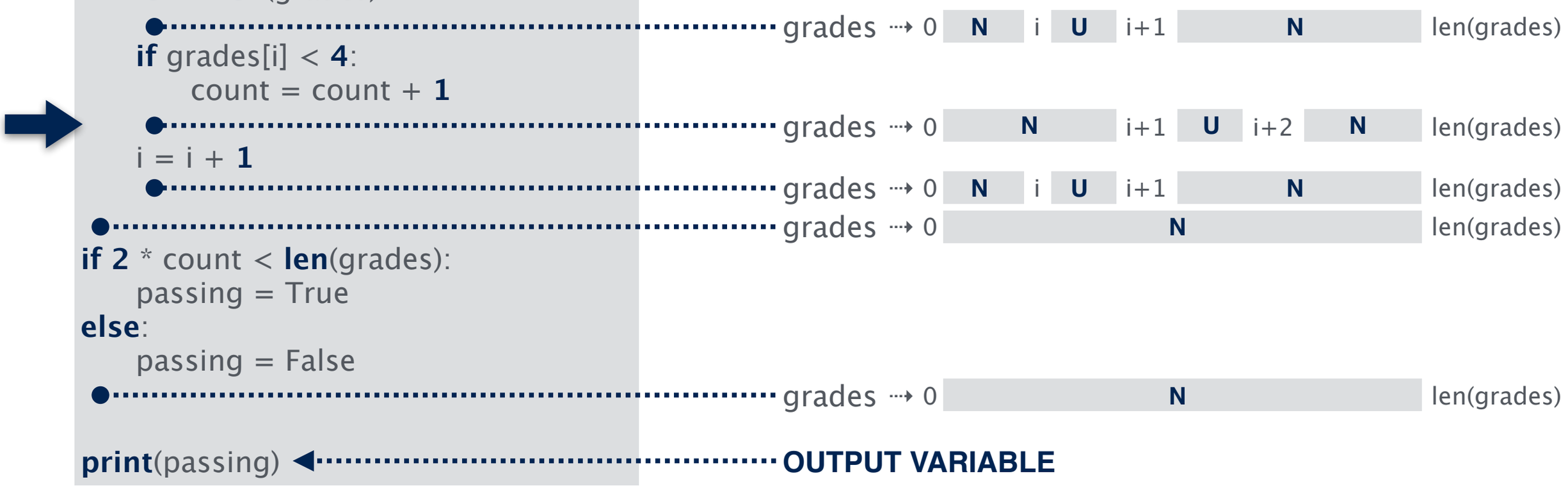
Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

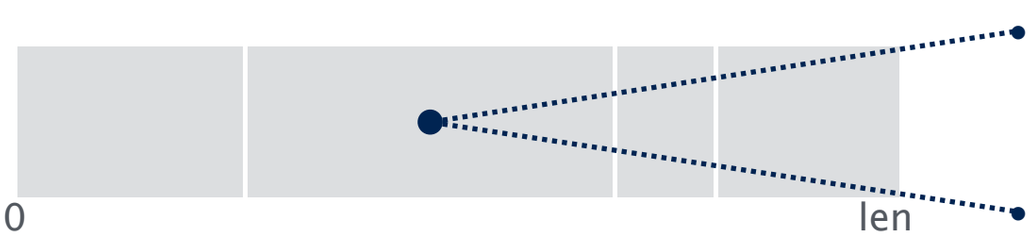
INPUT VARIABLE

ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

INPUT VARIABLE

ERROR: 1 SHOULD BE 0

grades ⇢ 0 | **N** | i | **U** | i+1 | **U** | i+2 | **N** | len(grades)

grades ⇢ 0 | **N** | i+1 | **U** | i+2 | **N** | len(grades)

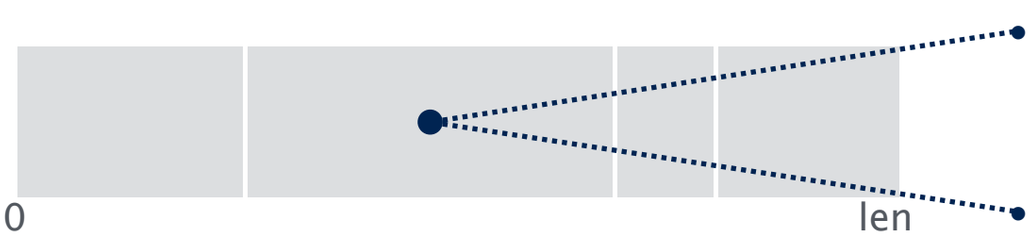
grades ⇢ 0 | **N** | i | **U** | i+1 | **N** | len(grades)

grades ⇢ 0 | **N** | len(grades)

grades ⇢ 0 | **N** | len(grades)

OUTPUT VARIABLE

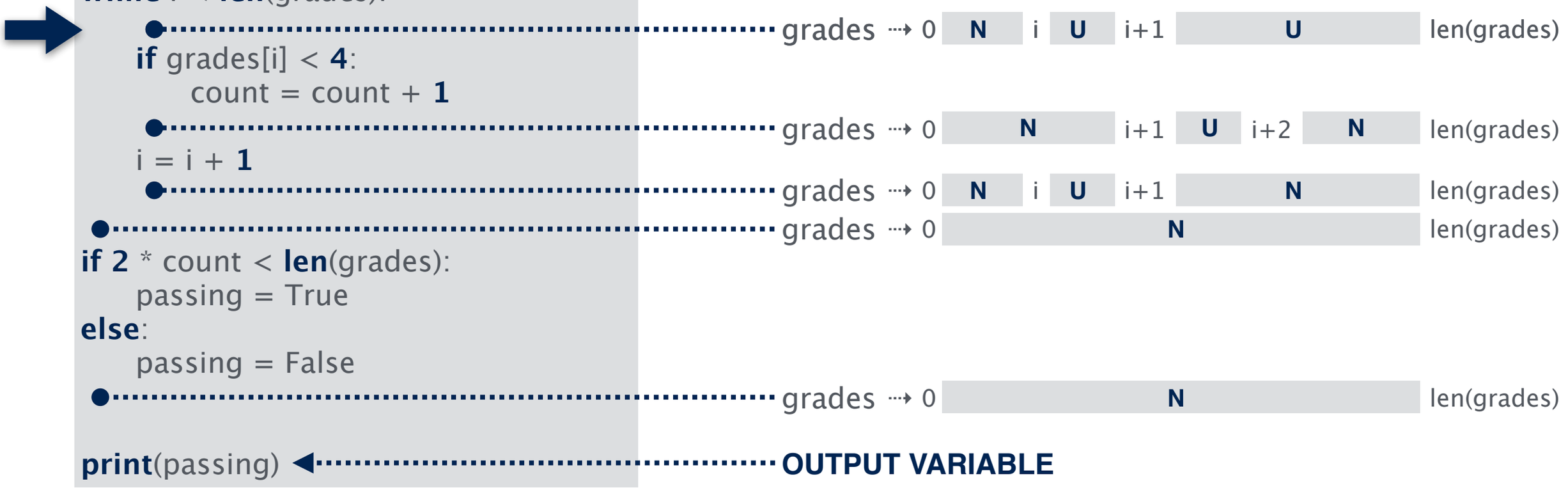
Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

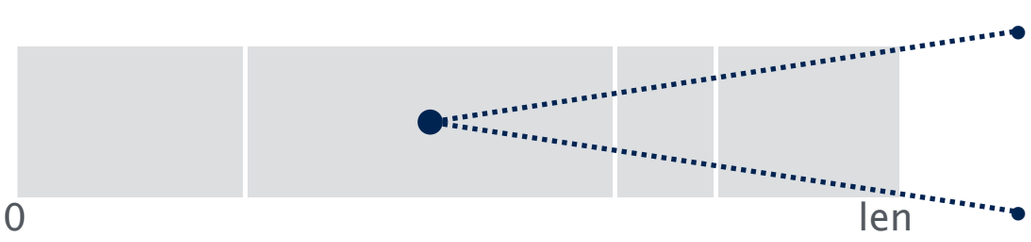
INPUT VARIABLE

ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

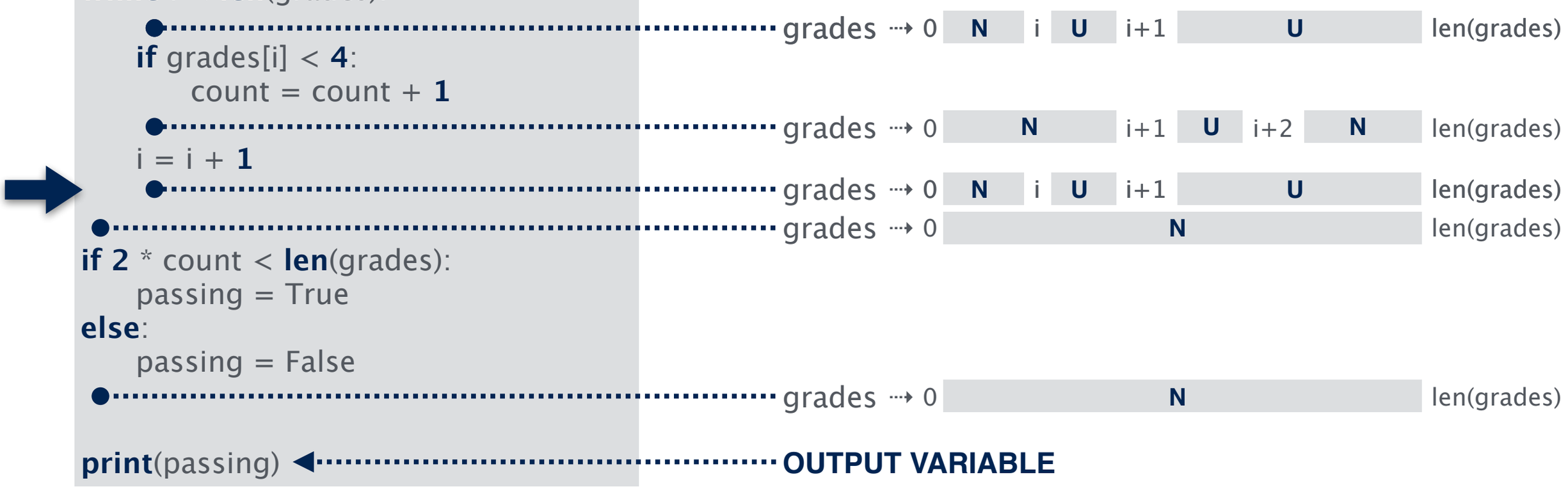
Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

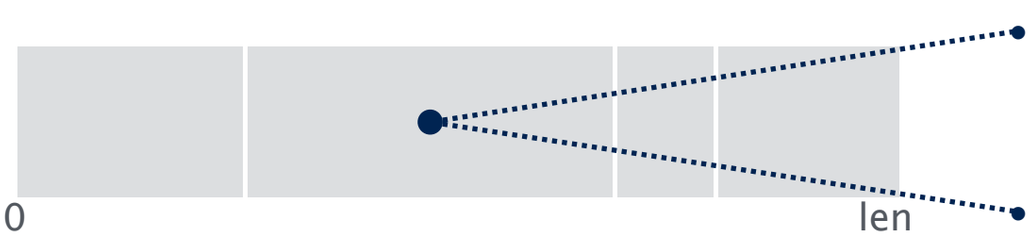
INPUT VARIABLE

ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

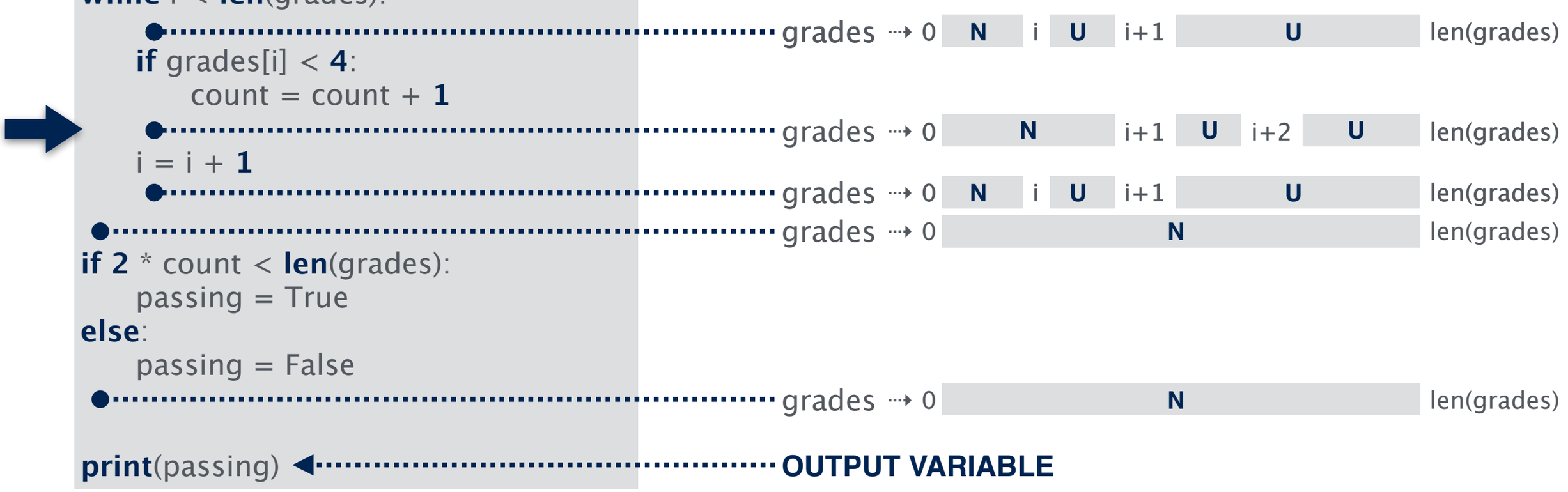
Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

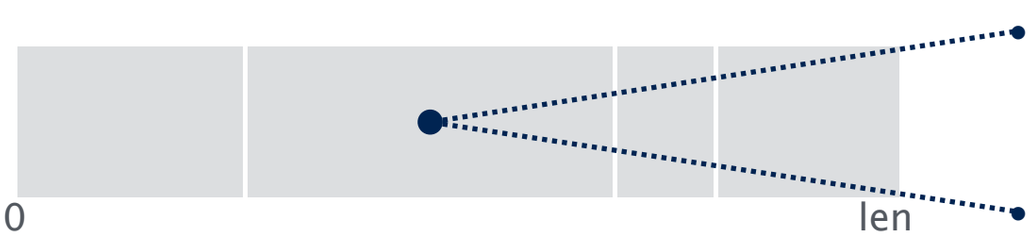
INPUT VARIABLE

ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

Piecewise Unused Input Data Analysis




```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
```

INPUT VARIABLE

ERROR: 1 SHOULD BE 0

grades ⇢ 0 **N** i **U** i+1 **U** len(grades)

grades ⇢ 0 **N** i+1 **U** i+2 **U** len(grades)

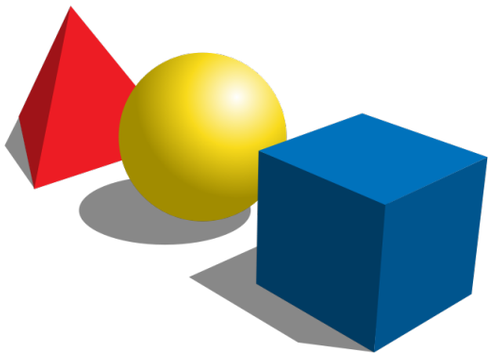
grades ⇢ 0 **N** i **U** i+1 **U** len(grades)

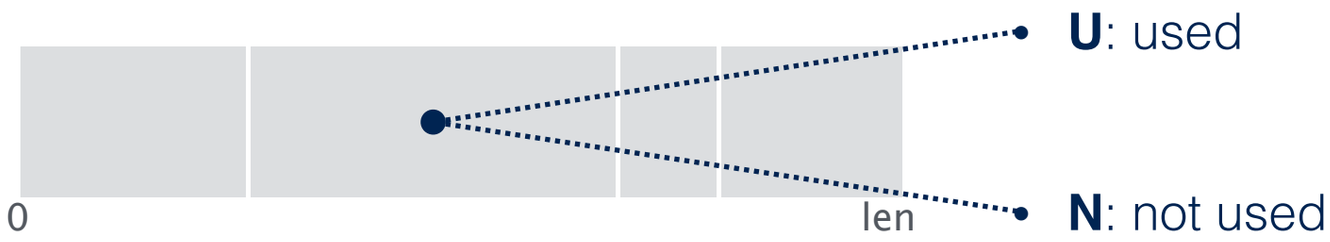
grades ⇢ 0 **N** len(grades)

grades ⇢ 0 **N** len(grades)

OUTPUT VARIABLE

Piecewise Unused Input Data Analysis

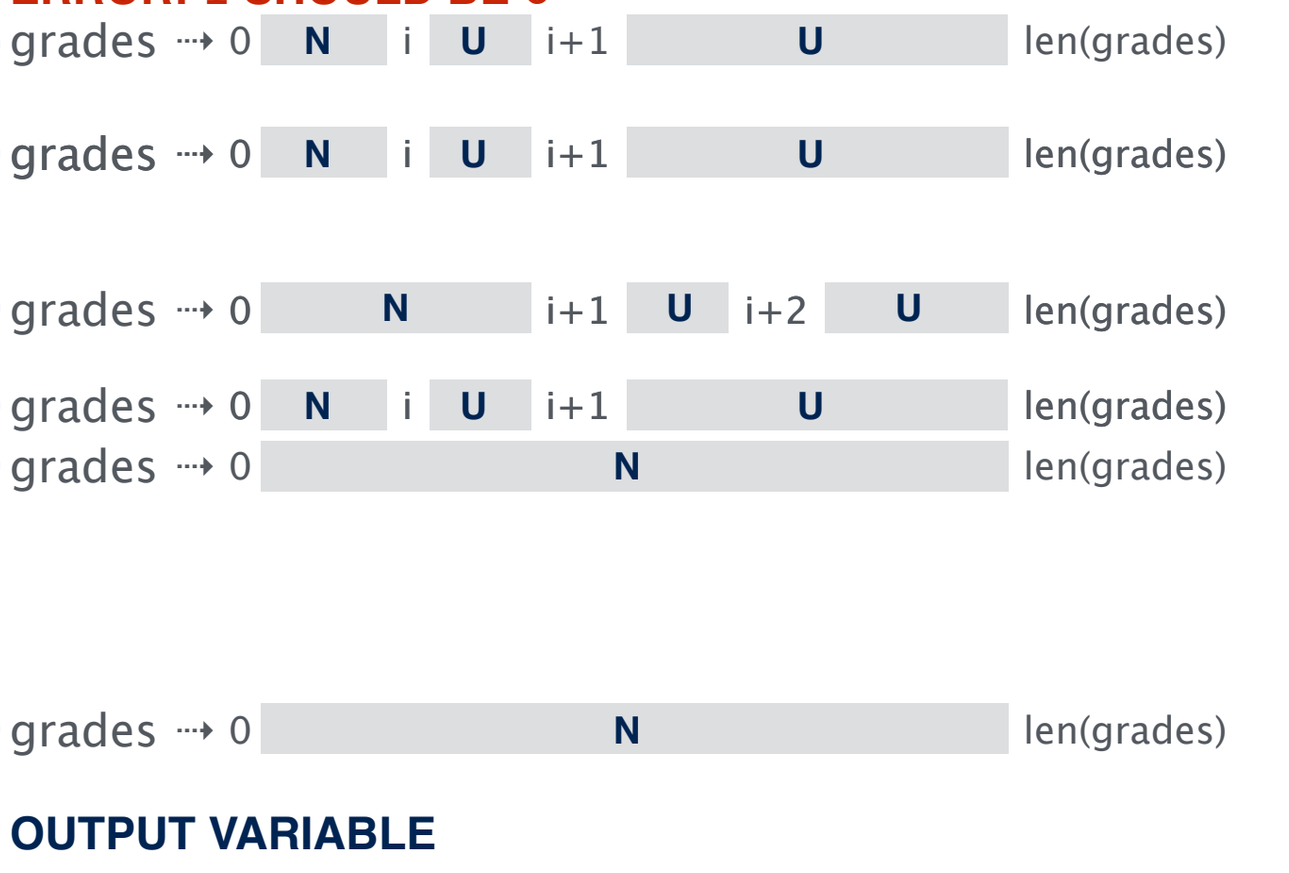





```
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
```

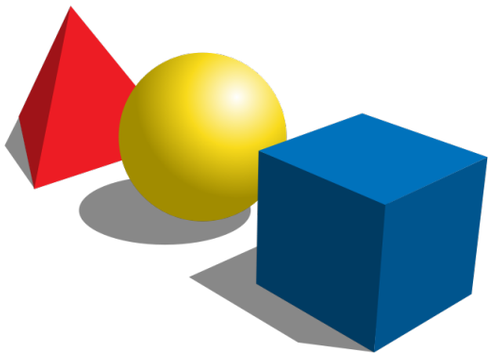
INPUT VARIABLE

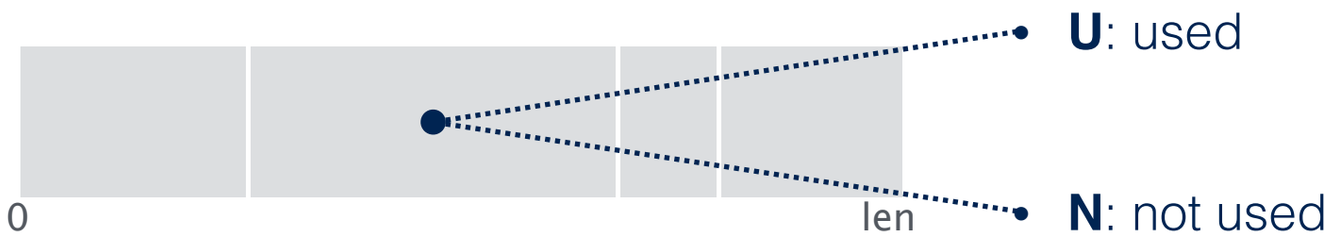
ERROR: 1 SHOULD BE 0



OUTPUT VARIABLE

Piecewise Unused Input Data Analysis






```
grades = list(map(int, input().split()))

count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
```

INPUT VARIABLE

grades ⇢ 0 **N** 1 **U** len(grades)

ERROR: 1 SHOULD BE 0

grades ⇢ 0 **N** i **U** i+1 **U** len(grades)

grades ⇢ 0 **N** i **U** i+1 **U** len(grades)

grades ⇢ 0 **N** i+1 **U** i+2 **U** len(grades)

grades ⇢ 0 **N** i **U** i+1 **U** len(grades)

grades ⇢ 0 **N** len(grades)

grades ⇢ 0 **N** len(grades)

OUTPUT VARIABLE

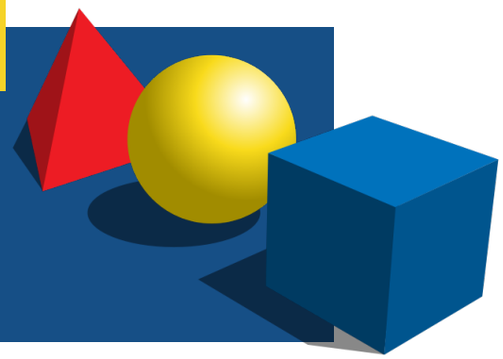
Unifying Framework

practical tools
targeting specific programs

secure information flow

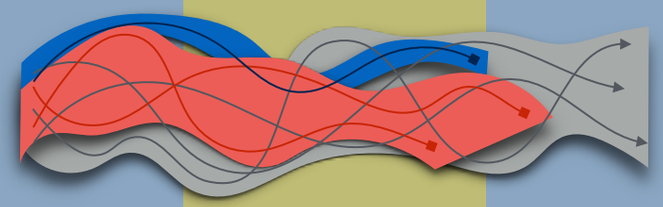
program slicing

algorithmic approaches
to decide program properties



strongly-live variable analysis

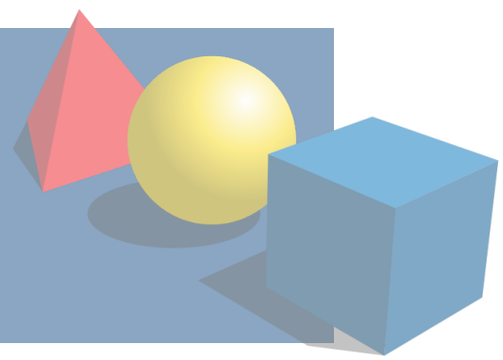
mathematical models
of the program behavior



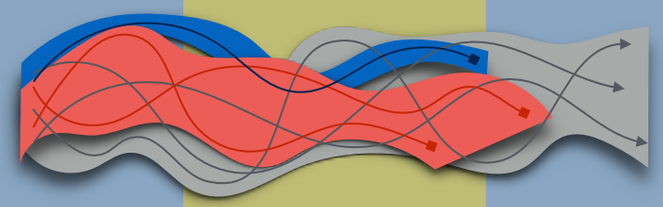
Lyra

practical tools
targeting specific programs

algorithmic approaches
to decide program properties



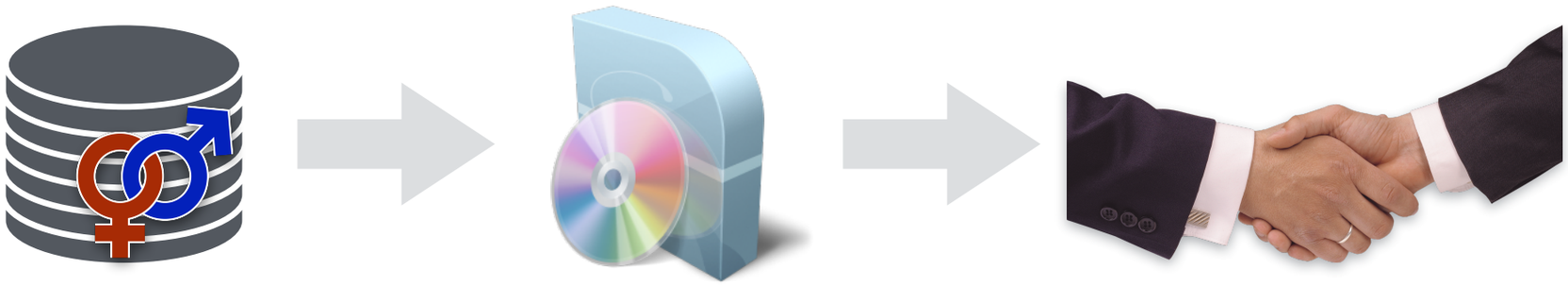
mathematical models
of the program behavior



Possible Applications

Data Provenance

Algorithmic Bias

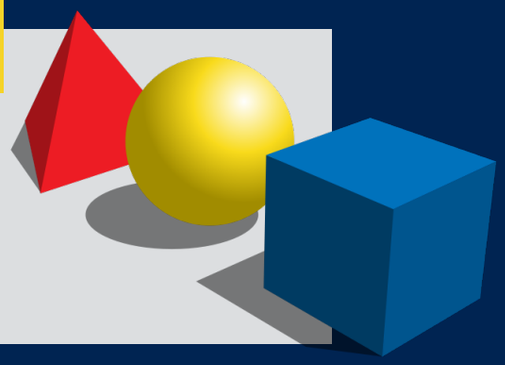


Data Privacy

practical tools
targeting specific programs

secure information flow

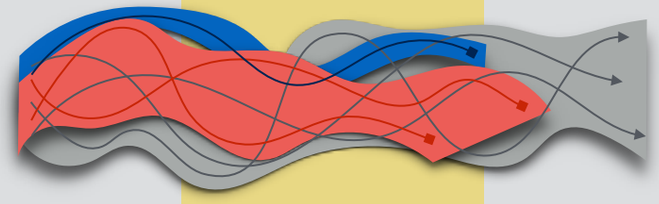
program slicing



algorithmic approaches
to decide program properties

strongly-live variable analysis

mathematical models
of the program behavior



QUESTIONS?