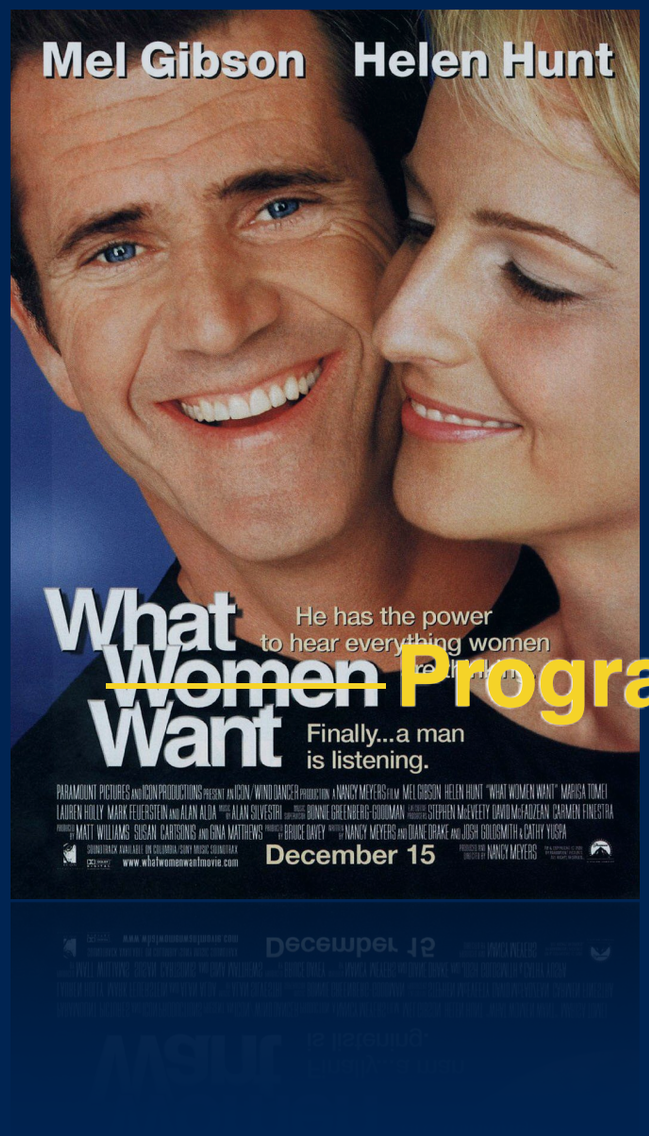
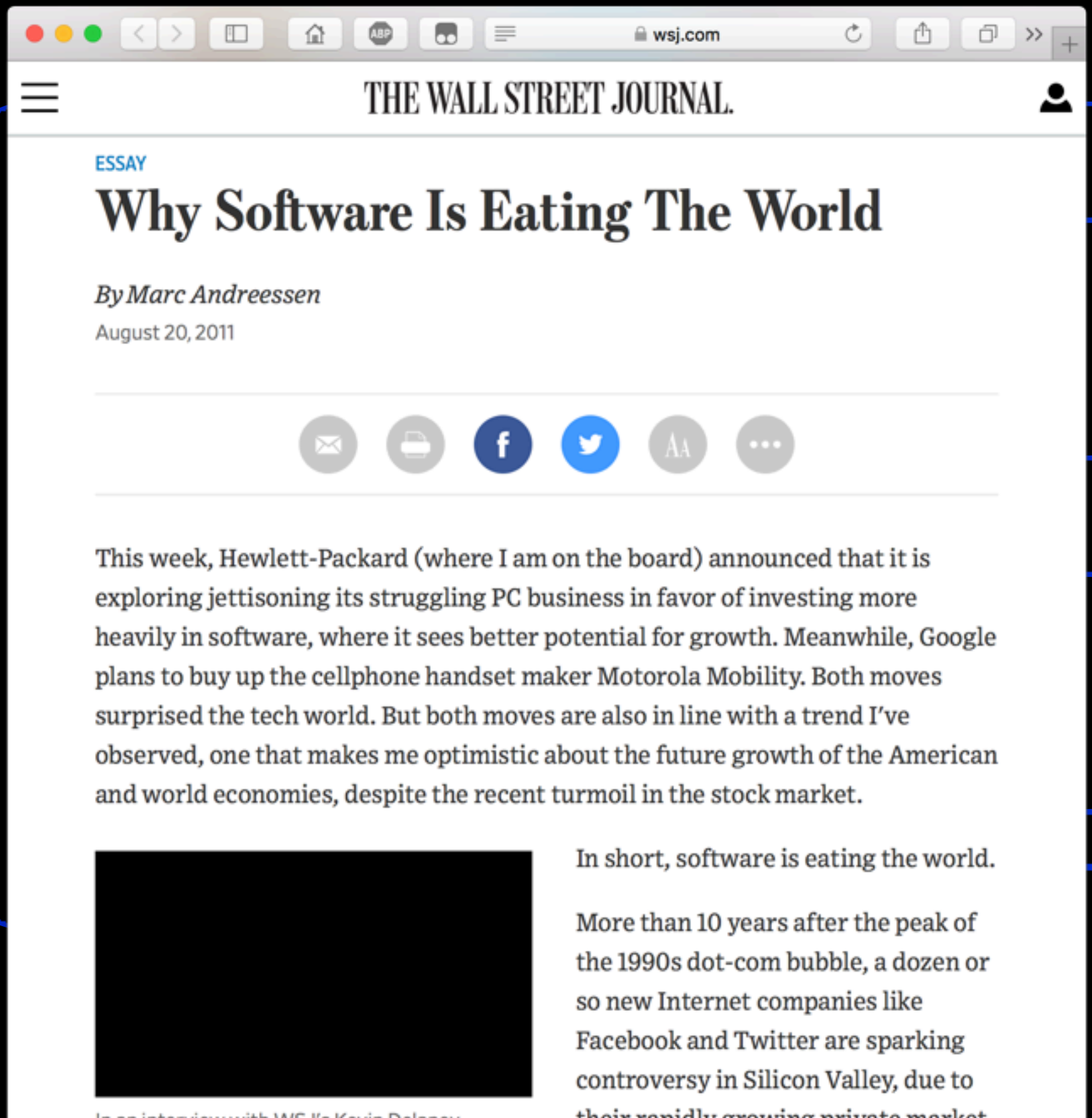


Automatic Inference of Input Data Specifications

~~Women~~ Programs

Caterina Urban
ANTIQUE Research Team
INRIA & École Normale Supérieure, Paris, France



ESSAY

Why Software Is Eating The World

By Marc Andreessen

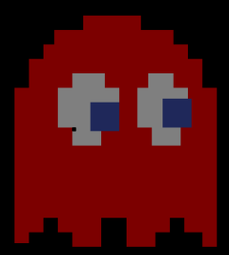
August 20, 2011

This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I've observed, one that makes me optimistic about the future growth of the American and world economies, despite the recent turmoil in the stock market.

In an interview with WSJ's Kevin Delaney,

In short, software is eating the world.

More than 10 years after the peak of the 1990s dot-com bubble, a dozen or so new Internet companies like Facebook and Twitter are sparking controversy in Silicon Valley, due to their rapidly growing private market



How Companies Learn Your Secrets

By CHARLES DUHIGG FEB. 16, 2012

Antonio Bolf

Andrew
when tw
to ask ar
pregnan

Pole has
been obs
his life.
were goi
"The ste
last year

As the m

Algorithm Helps Send ison

any insight, to play a role in sending a man to prison?

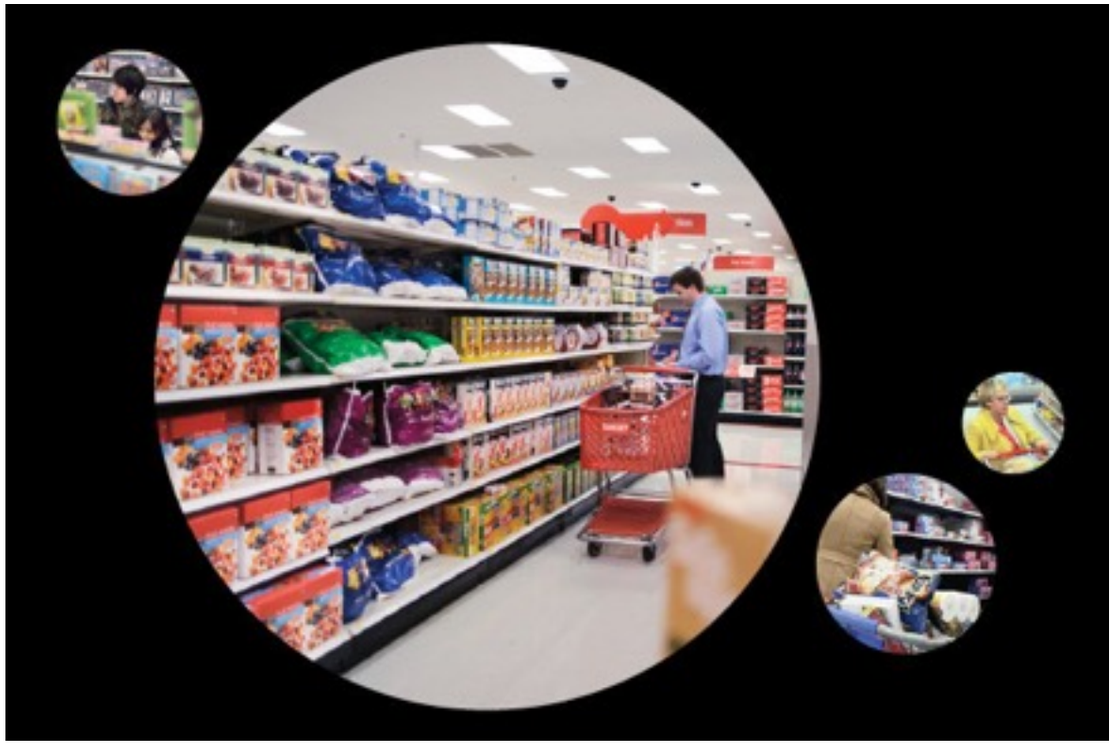
Wisconsin is one of several states using algorithms in the sentencing process. Above, the state capitol.Michael P. King/Wisconsin State Journal, via Associated Press

software is **automating processes**

software is **helping us make decisions**

How Companies Learn Your Secrets

By CHARLES DUHIGG FEB. 16, 2012



Antonio Bolfo/Reportage for The New York Times

Andrew Pole had just started working as a statistician for Target in 2002, when two colleagues from the marketing department stopped by his desk to ask an odd question: "If we wanted to figure out if a customer is pregnant, even if she didn't want us to know, can you do that? "

Pole has a master's degree in statistics and another in economics, and has been obsessed with the intersection of data and human behavior most of his life. His parents were teachers in North Dakota, and while other kids were going to 4-H, Pole was doing algebra and writing computer programs. "The stereotype of a math nerd is true," he told me when I spoke with him last year. "I kind of like going out and evangelizing analytics."

privacy violations

When an Algorithm Helps Send You to Prison

Oct. 26, 2017

Op-Ed Contributor

In 2013, police officers in Wisconsin arrested a man driving a car that had been used in a recent shooting. The man, Eric Loomis, pleaded guilty to attempting to flee an officer, and no contest to operating a vehicle without the owner's consent. Neither of his crimes mandates prison time.

At Mr. Loomis's sentencing, the judge cited, among other factors, Mr. Loomis's high risk of recidivism as predicted by a computer program called COMPAS, a risk assessment algorithm used by the state of Wisconsin. The judge denied probation and prescribed an 11-year sentence: six years in prison, plus five years of extended supervision.

N

unfair decisions

Mr. Loomis challenged the use of an algorithm as a violation of his due process rights to be sentenced individually, and without consideration of impermissible factors like gender. The Wisconsin Supreme Court rejected his challenge. In June, the United States Supreme Court declined to hear his case, meaning a majority of justices effectively condoned the algorithm's use. Their decision will have far-ranging effects.

Something about this story is fundamentally wrong: Why are we allowing a computer program, into which no one in the criminal justice system has any insight, to play a role in sending a man to prison?

Wisconsin is one of several states using algorithms in the sentencing process. Above, the state capitol.Michael P. King/Wisconsin State Journal, via Associated Press

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

incorrect results

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

Implicit Assumptions

```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

```
[^\n ] [1-9][0-9]* (A|B|C|D|F){ }
[^\n ] [1-9][0-9]* (A|B|C|D|F){   }
[^\n ] [1-9][0-9]* (A|B|C|D|F){   }
[^\n ] [1-9][0-9]* (A|B|C|D|F){   }
[^\n ] [1-9][0-9]* (A|B|C|D|F){   }
[^\n ] [1-9][0-9]* (A|B|C|D|F){   }
[^\n ] [1-9][0-9]* (A|B|C|D|F){   }
```

Examples

- each line contains a certain number of characters or words
- all characters are uppercase or lowercase
- values can only be in a certain range

```
dkd 2 A C
ndd 1 F
dle 3 C C C
bk2 1 B
wwb 2 D F
wbd 1 D
```

```
dkd 2 A C
ndd 1 F
dle 3.0 C C C
bk2 1 B
wwb 2 D F
wbd 1 D
```

```
dkd 2 A C
ndd 1 F
dle 3 C C C
bk2 1 B+
wwb 2 D F
wbd 1 D
```

```
dkd 2 A C
ndd 1 F
dle 3 C C C
bk2 1 B
wwb 2 D
wbd 1 D
```

```
dkd 1 A C
ndd 1 F
dle 3 C C C
bk2 1 B
wwb 1 D
wbd 2 D F
```

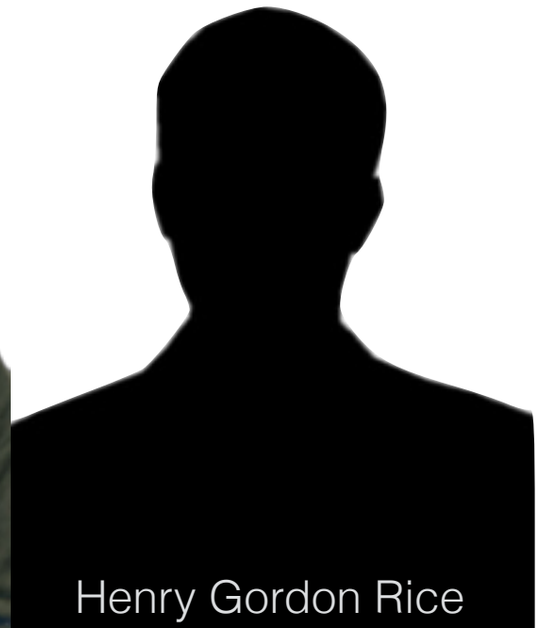
Our (Ideal) Solution


```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0,
'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data =
line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades +
2):
            gpa +=
grade2gpa[data[i]]
        result = gpa / grades

        print('{}:
{}'.format(data[0], result))
```

program

input data
specification


Alan Turing


Henry Gordon Rice

Sufficient and Necessary Conditions

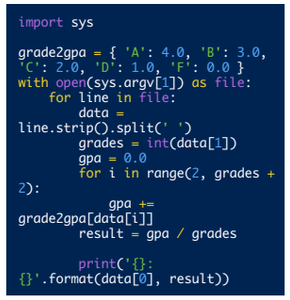
✔

false alarm

✘

false positive

Our (Actual) Solution



program

static program analysis

input data specification

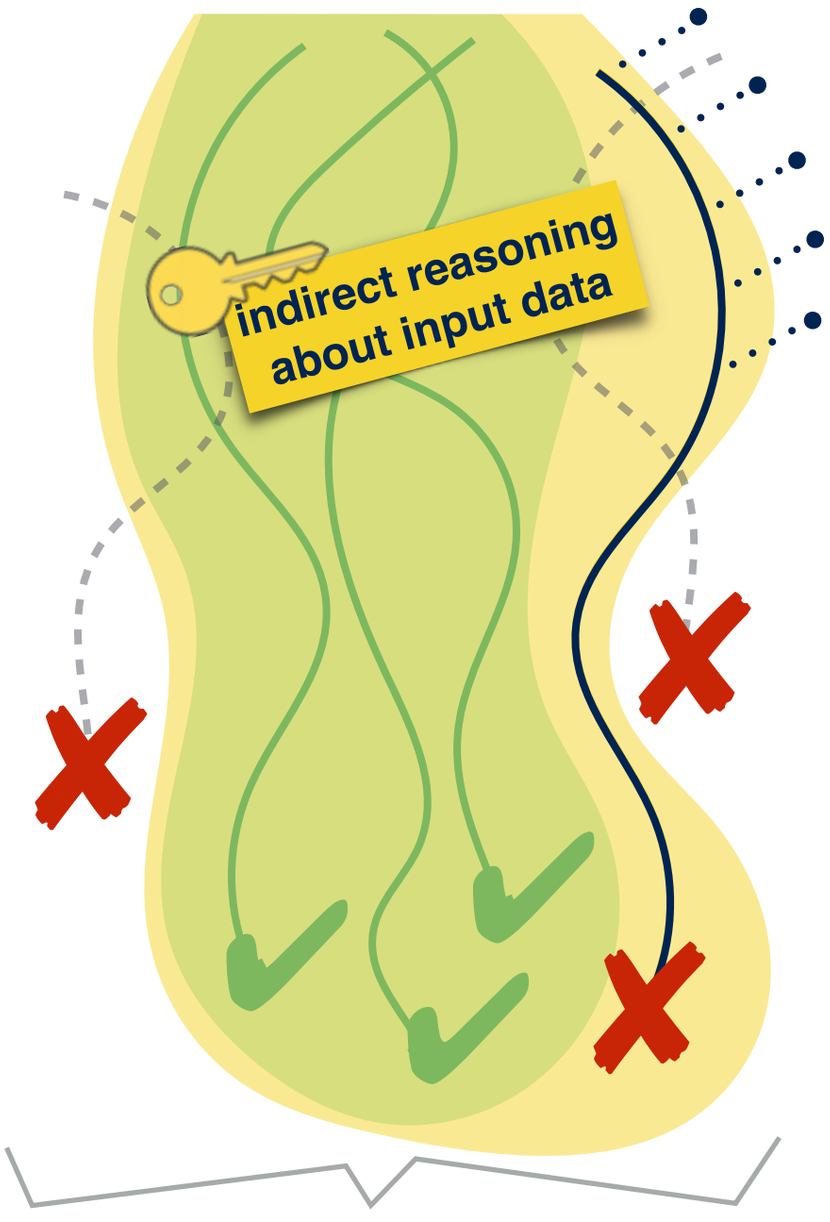
data

data cleaning

data

data checking

Static Analysis in Theory




```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0,
'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data =
line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades +
2):
            gpa +=
grade2gpa[data[i]]
        result = gpa / grades

        print('{}:
{}'.format(data[0], result))
```

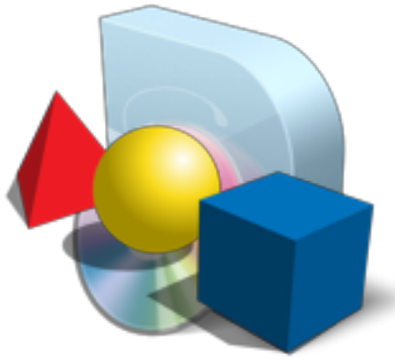
Property
all program executions that are *correct*

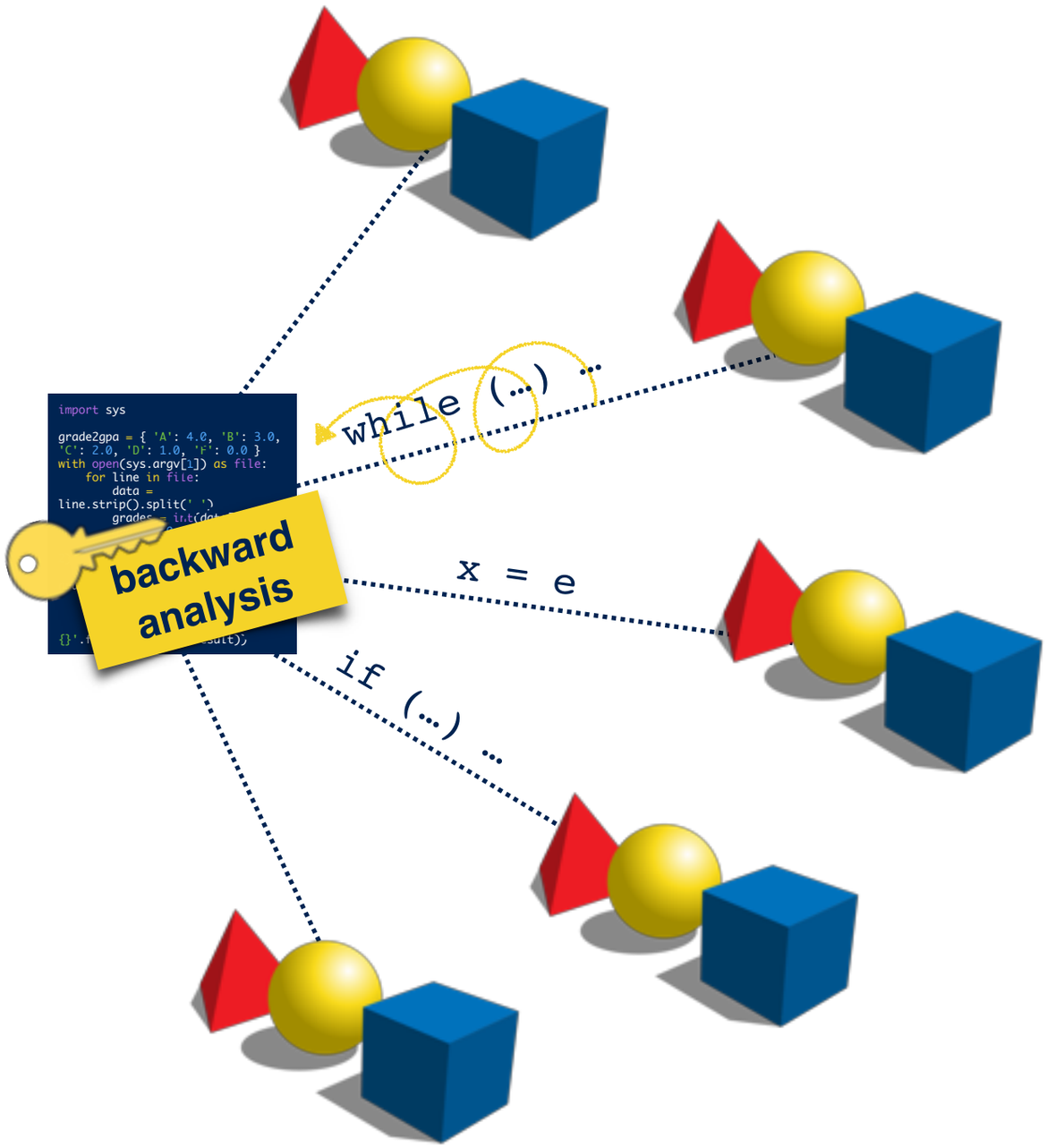
Example: all executions that do not crash the program

Abstraction
all program executions that *might be correct*

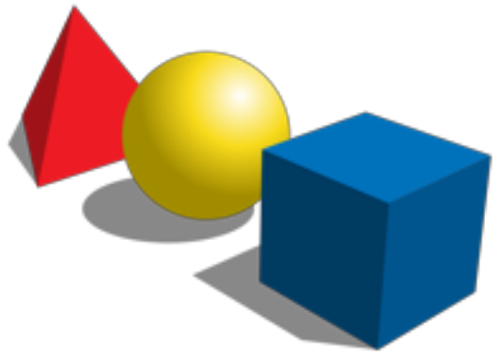
Example: all executions that do not crash the program
plus maybe some others that do

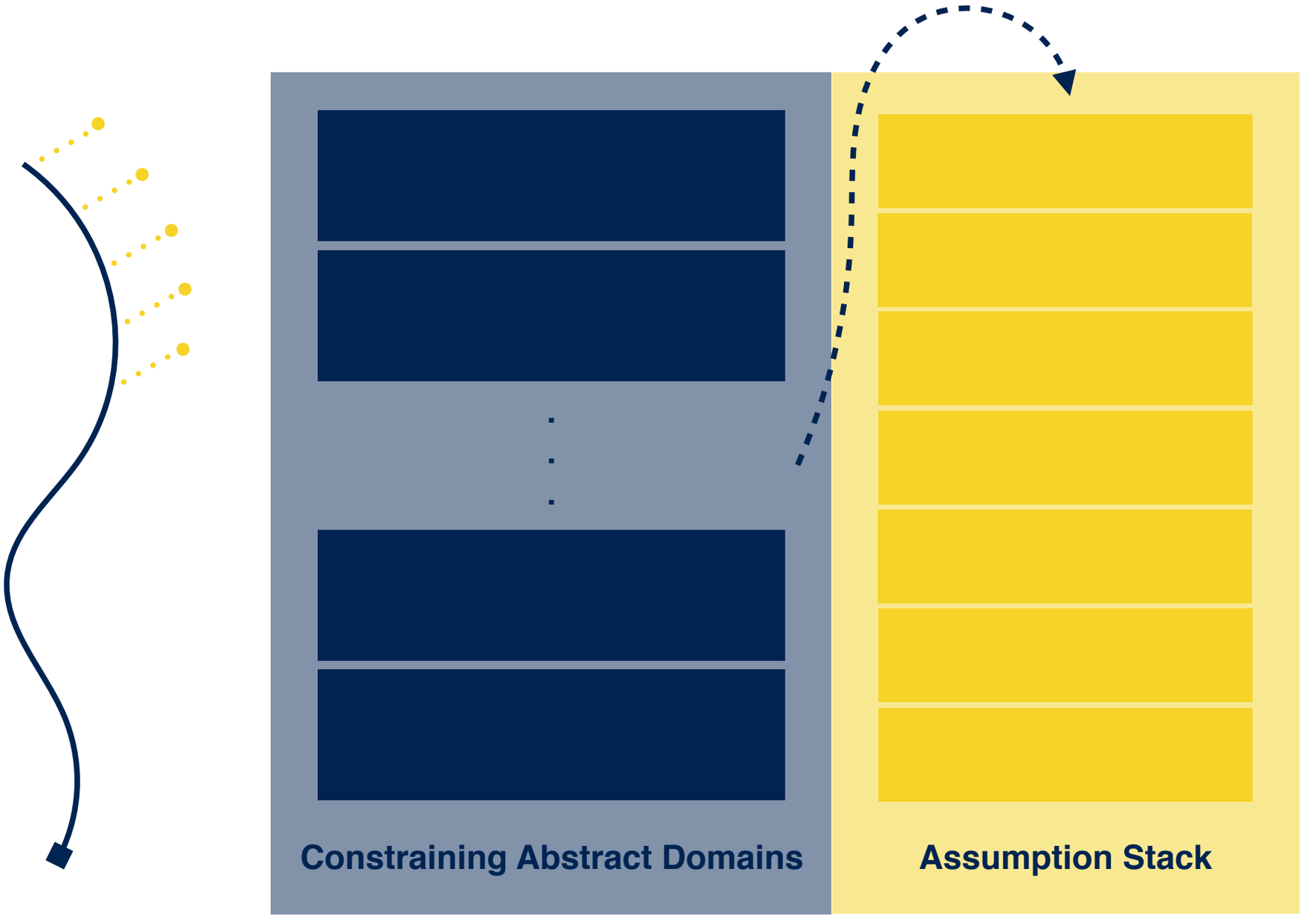
Static Analysis in Practice



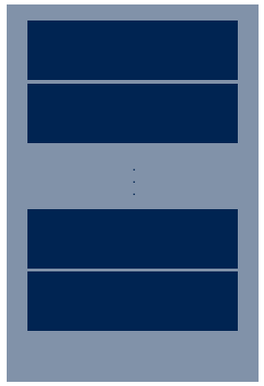


Implicit Assumption Abstract Domain



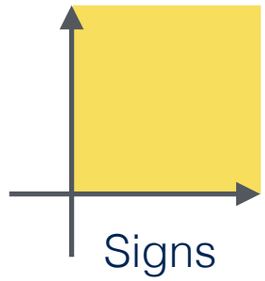


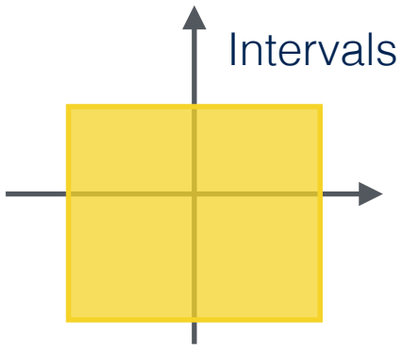
Constraining Abstract Domains

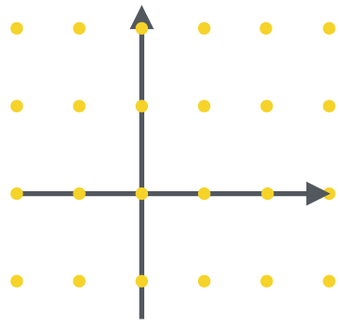


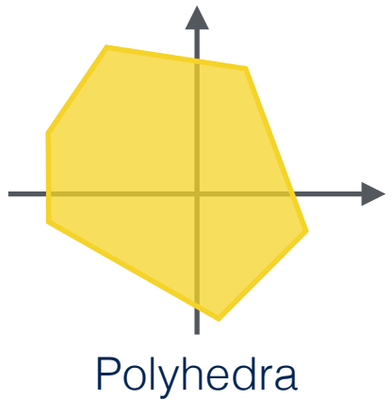
Type Abstract Domain

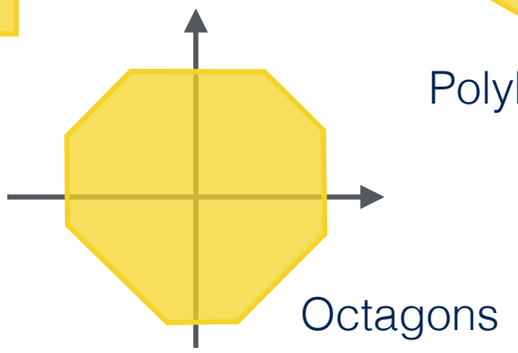
Numerical Abstract Domains



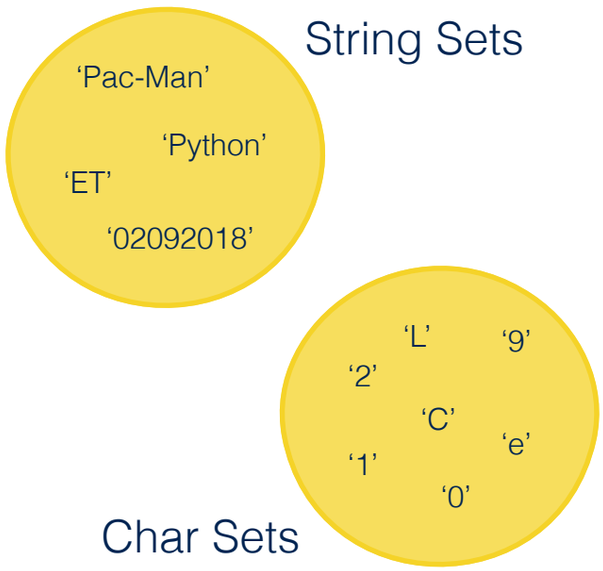








String Abstract Domains



Collection Abstract Domains

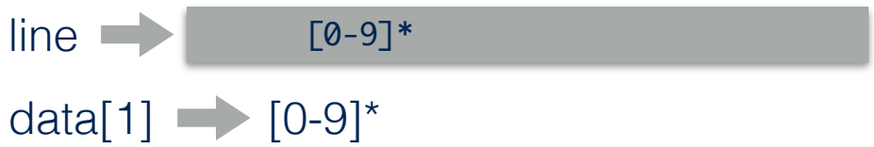
Type Abstract Domain


```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

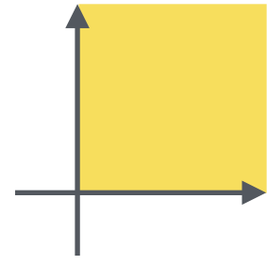
data[1] ➡ [0-9]*

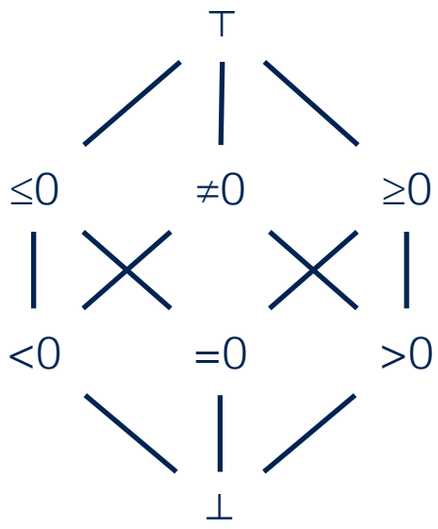
Type Abstract Domain


```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```



Sign Abstract Domain



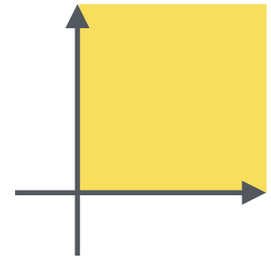


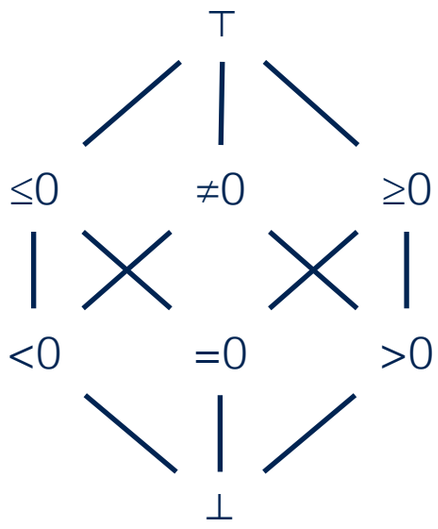
var1 ➡ ≥0
var2 ➡ <0
… ➡ …

```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

grades ➡ ≠0

Sign Abstract Domain





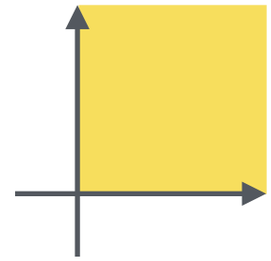
var1 → ≥0
var2 → <0
… → …

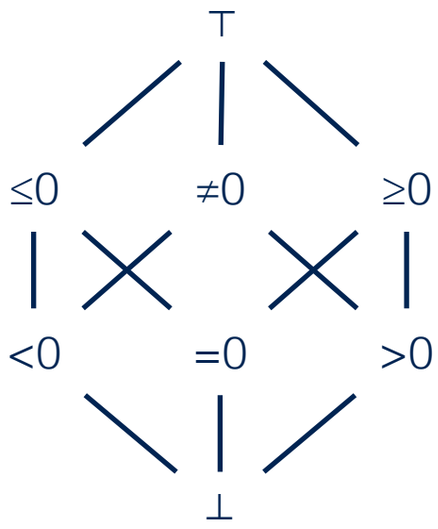
```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

data[1] → [1-9][0-9]*

grades → ≠0

Sign Abstract Domain





var1 → ≥0
var2 → <0
… → …

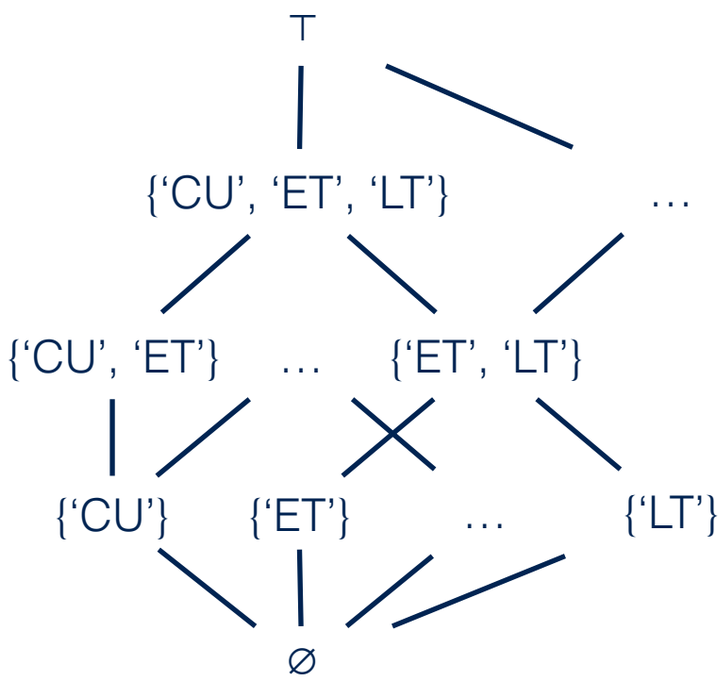
```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

line → `[1-9][0-9]*`

data[1] → [1-9][0-9]*

grades → ≠0

String Set Abstract Domain

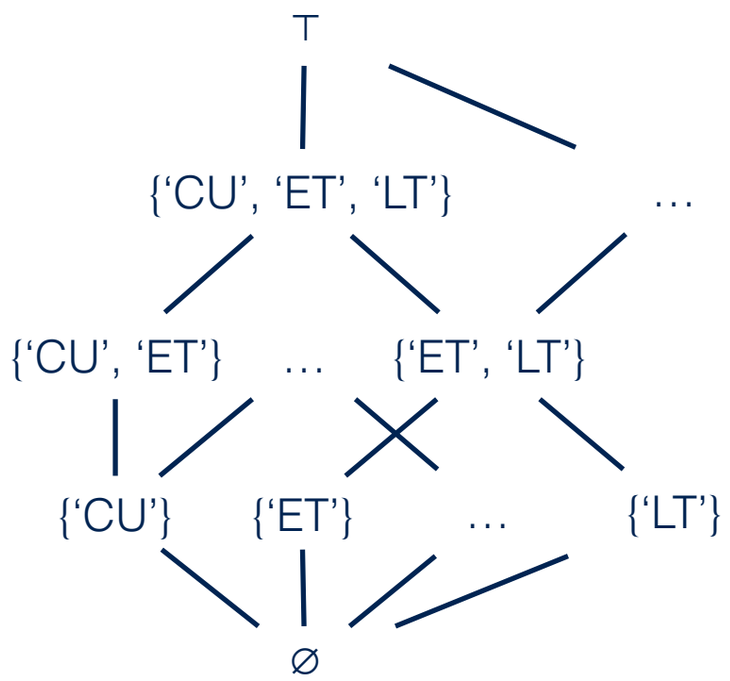


var1 → {‘CU’}
var2 → {‘ET’, ‘LT’}
… → …

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

data[i] → {‘A’, ‘B’, ‘C’, ‘D’, ‘F’}

String Set Abstract Domain

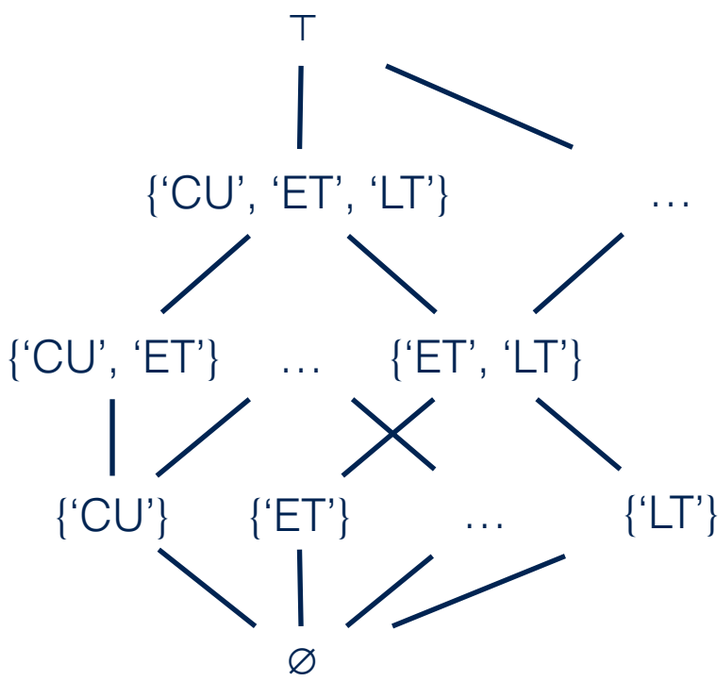


var1 ➡ {‘CU’}
var2 ➡ {‘ET’, ‘LT’}
… ➡ …

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

data[2…grades+2] ➡ {‘A’, ‘B’, ‘C’, ‘D’, ‘F’}
data[i] ➡ {‘A’, ‘B’, ‘C’, ‘D’, ‘F’}

String Set Abstract Domain



var1 ➡ {‘CU’}
var2 ➡ {‘ET’, ‘LT’}
… ➡ …

```python
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

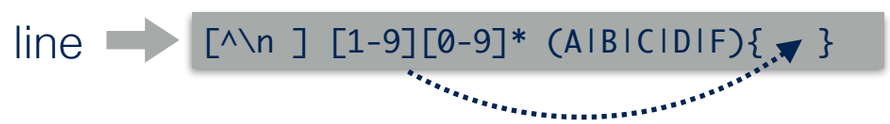
line ➡ `[1-9][0-9]*  (A|B|C|D|F){ }`

data[2…grades+2] ➡ {‘A’, ‘B’, ‘C’, ‘D’, ‘F’}

data[i] ➡ {‘A’, ‘B’, ‘C’, ‘D’, ‘F’}

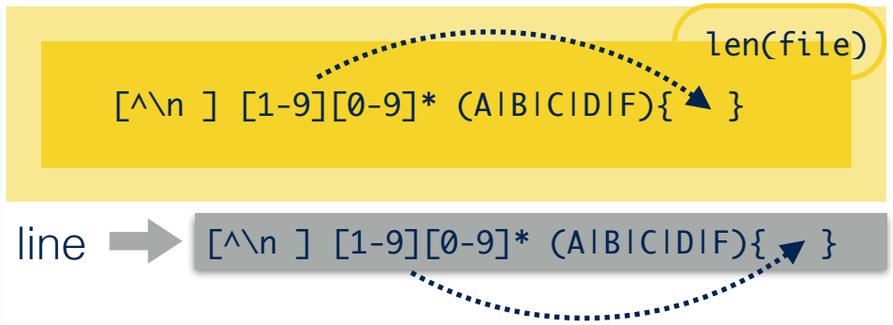
Assumption Stack

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```



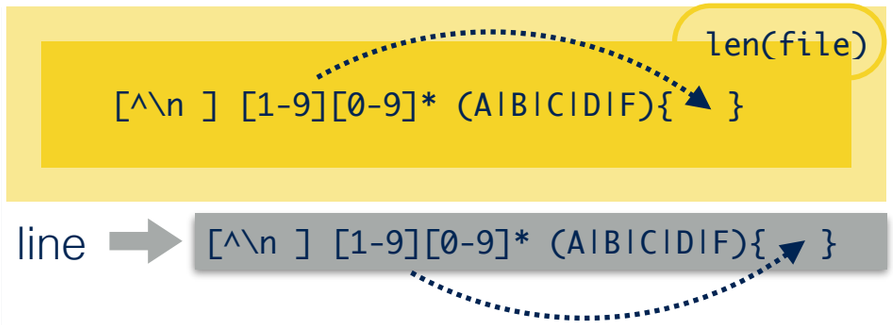
Assumption Stack

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```



Assumption Stack

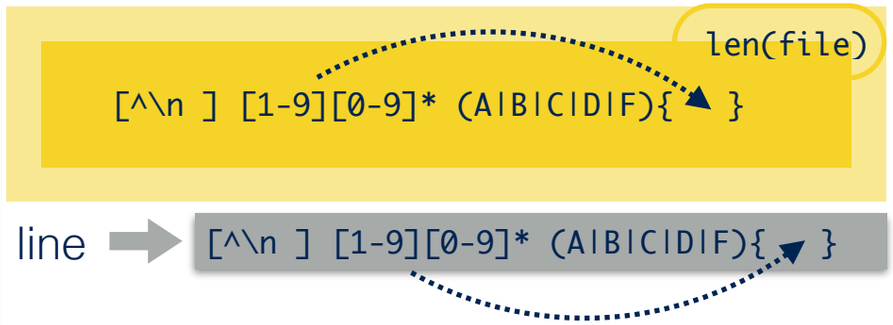
```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```



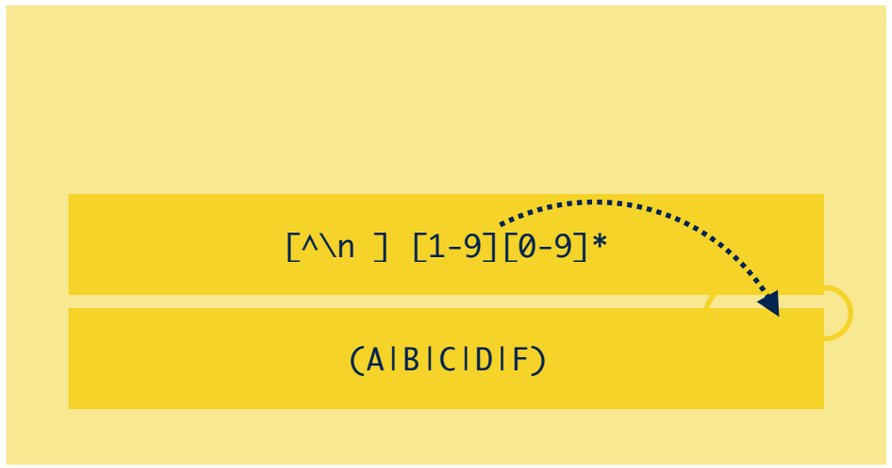

```
grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }

students = int(input())
for _ in range(students):
    data = input().split(' ')
    grades = int(data[1])
    gpa = 0.0
    for _ in range(grades):
        gpa += grade2gpa[input()]

    print('{}: {}'.format(data[0], gpa / grades))
```

Assumption Stack

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

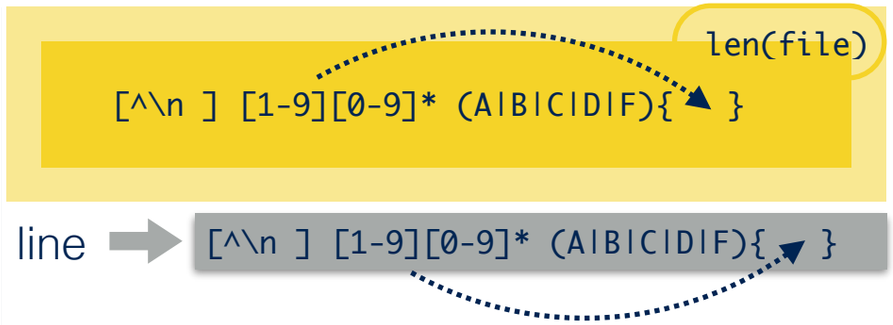



```
grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }

students = int(input())
for _ in range(students):
    data = input().split(' ')
    grades = int(data[1])
    gpa = 0.0
    for _ in range(grades):
        gpa += grade2gpa[input()]

    print('{}: {}'.format(data[0], gpa / grades))
```

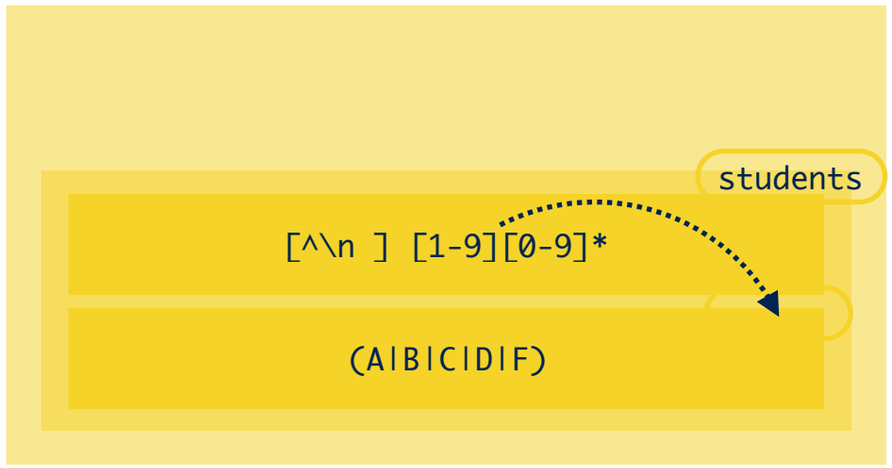


Assumption Stack

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```

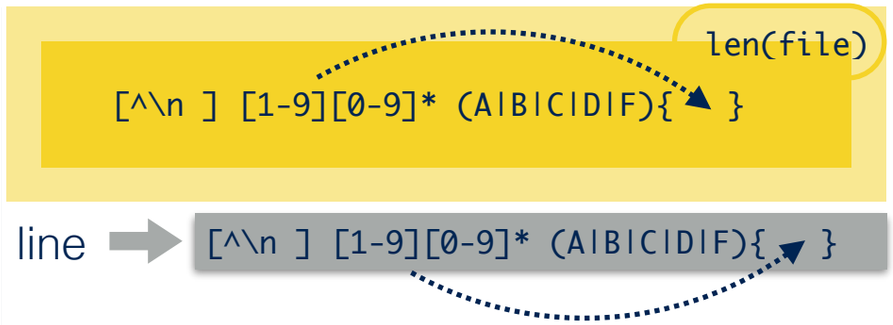



```
grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }

students = int(input())
for _ in range(students):
    data = input().split(' ')
    grades = int(data[1])
    gpa = 0.0
    for _ in range(grades):
        gpa += grade2gpa[input()]

    print('{}: {}'.format(data[0], gpa / grades))
```

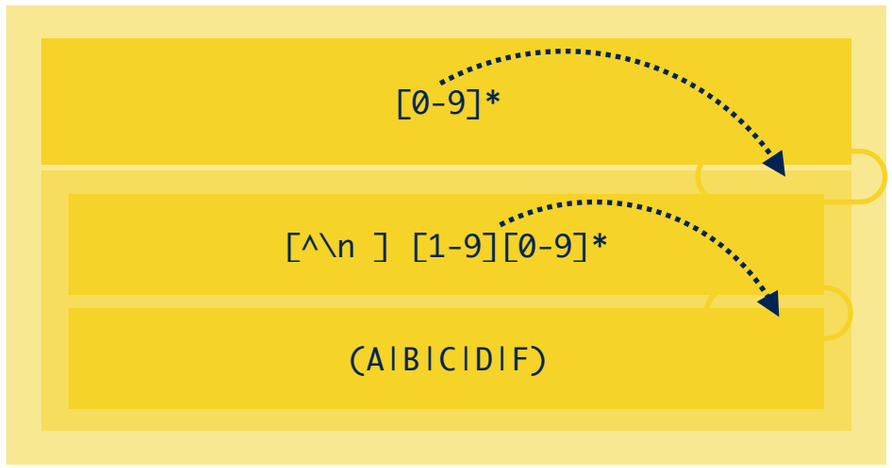


Assumption Stack

```
import sys

grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
with open(sys.argv[1]) as file:
    for line in file:
        data = line.strip().split(' ')
        grades = int(data[1])
        gpa = 0.0
        for i in range(2, grades + 2):
            gpa += grade2gpa[data[i]]
        result = gpa / grades

        print('{}: {}'.format(data[0], result))
```




```
grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }

students = int(input())
for _ in range(students):
    data = input().split(' ')
    grades = int(data[1])
    gpa = 0.0
    for _ in range(grades):
        gpa += grade2gpa[input()]

    print('{}: {}'.format(data[0], gpa / grades))
```



Implementation

https://github.com/caterinaurban/Lyra

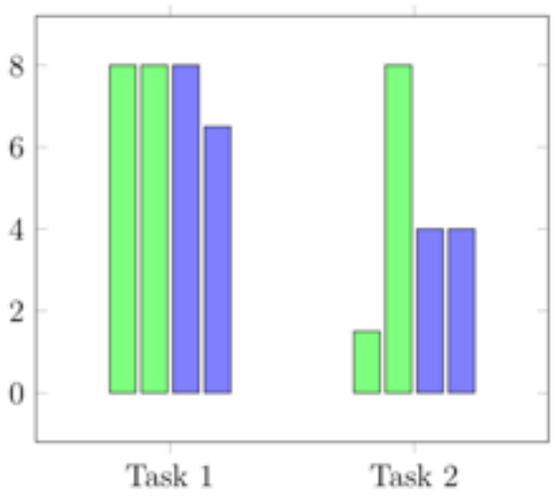
	Inputs	Soundness	Precision	Source of Imprecision		
				Type	Octagon	CharSet
if.py	2	✔	✔			
add.py	2	✔	✔			
cast.py	1	✔	✔			
assume.py	1	✔	✘(1)			1
basic.py	1	✔	✘(1)			1
concatenation.py	2	✔	✘(2)			2
must.py	1	✔	✘(1)			1
substitution.py	3	✔	✘(3)			3
cars.py	5	✔	✘(1)		1	
convert.py	4	✔	✘(2)		1	1
dna.py	5	✔	✘(2)			2
food.py	5	✔	✘(1)	1		
grades.py	8	✔	✘(1)			1
str.py	2	✔	✘(2)			2
triangle.py	3	✔	✘(3)		3	
type.py	4	✔	✘(1)			1
unification.py	3	✔	✘(2)	1	2	
123.py	2	✔	✔			
branch.py	3	✔	✘(1)	1		
filter.py	1	✔	✔			
five.py	3	✔	✘(1)			1
merge.py	3	✔	✔			

Data Checking

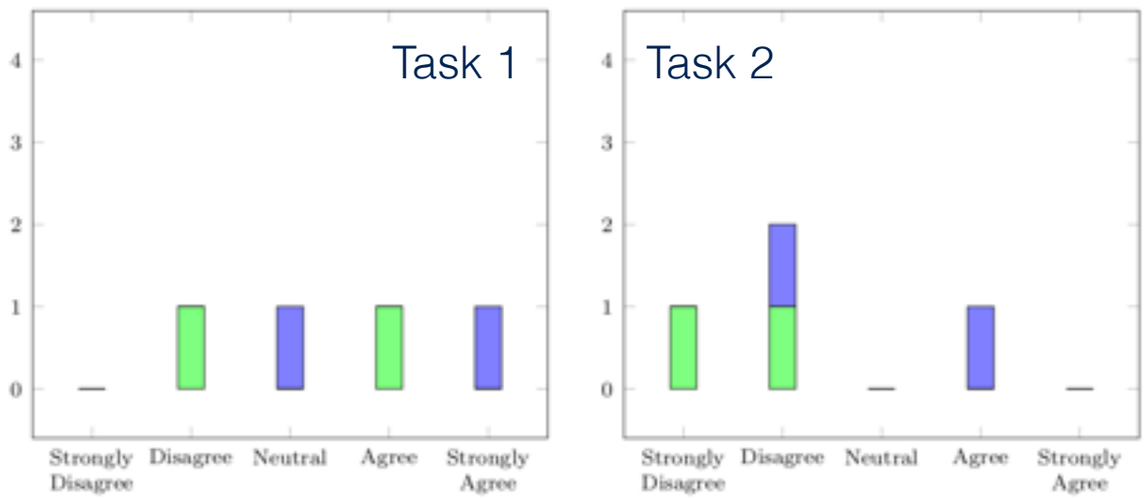
Stand-alone Tool

User Study

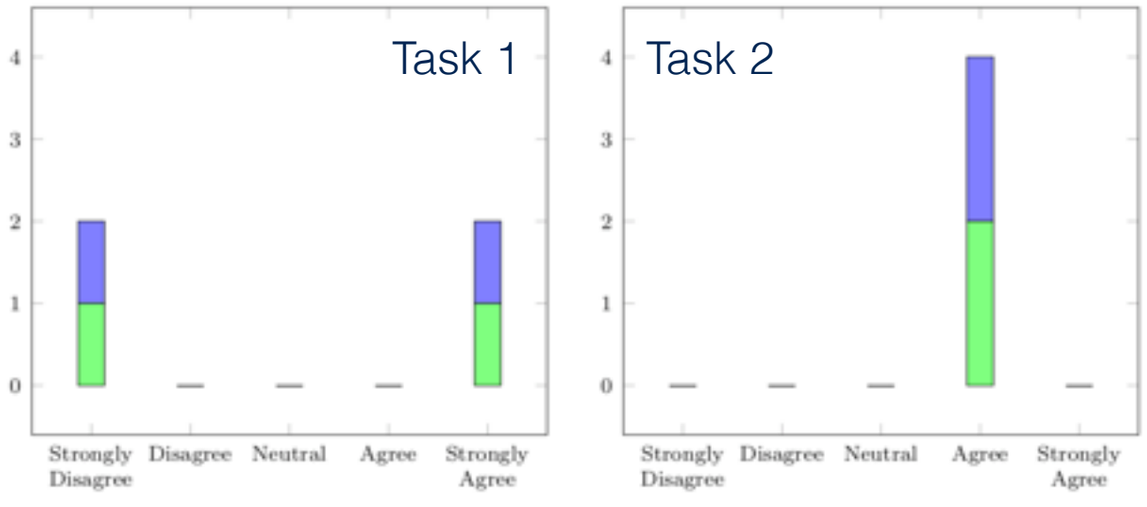
- 4 participants (2 **with** and 2 **without** computer science background)
- 2 tasks (without — task 1 — and with — task 2 — the help of the data checker)
- 8 minutes to solve each task



Time spent on each task



”The task is frustrating”



”The task is easy”

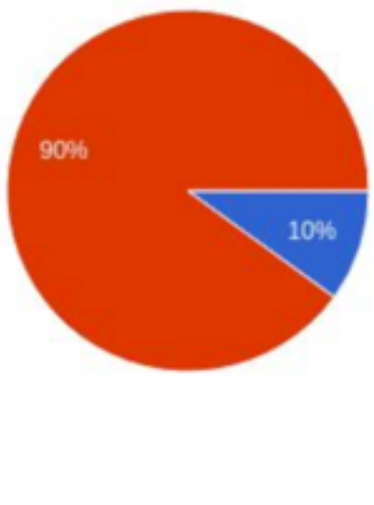
Data Checking

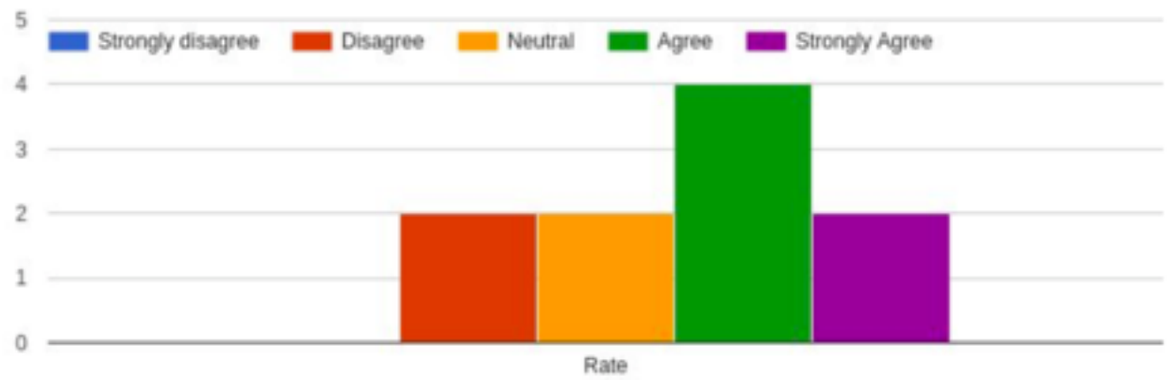
Text Editor Plugin

User Study

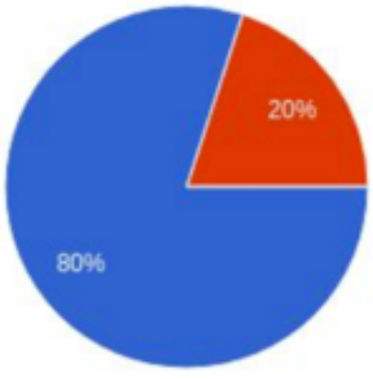
- 10 participants (5 with and 5 without computer science background)
- 2 tasks (without — task 1 — and with — task 2 — the help of the data checker)
- 8 minutes to solve each task

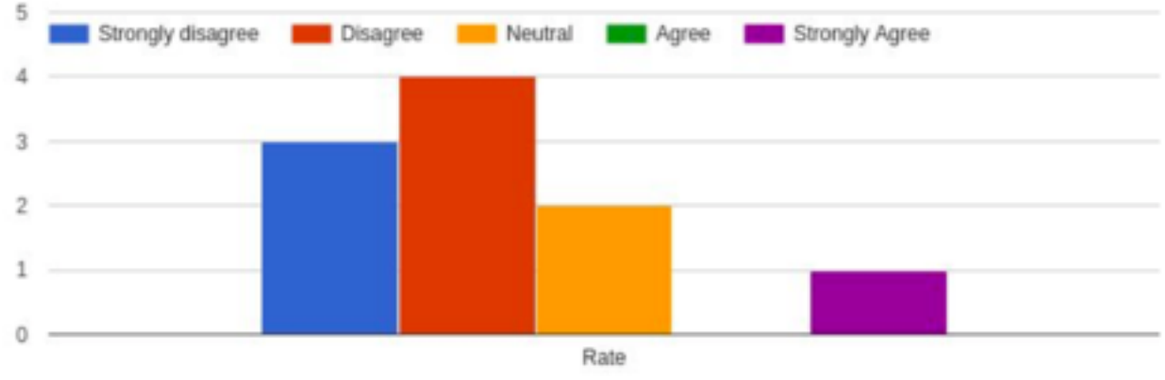
Task 1





Task 2





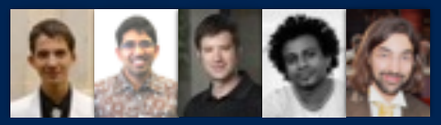
“I was able to complete the task”

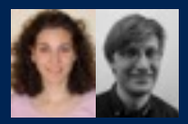
”The task is frustrating”

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

mathematical models
of the program behavior





practical tools
targeting specific programs

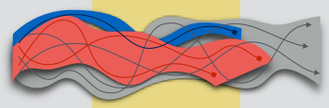
algorithmic approaches
to decide program properties

secure information flow

program slicing

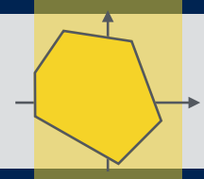
strongly-live variable analysis

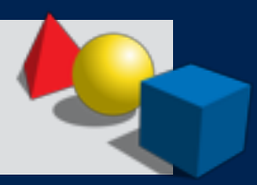
mathematical models
of the program behavior



practical tools
targeting specific programs

algorithmic approaches
to decide program properties





mathematical models
of the program behavior