

Learn to Build Automated Software Analysis Tools with Graph Paradigm and Interactive Visual Framework

ASE Tutorial, September 4, 2016

Suresh C. Kothari

Richardson Professor

Department of Electrical and Computer Engineering

Ben Holland, Iowa State University

Why abstract software?

Acknowledgment: Team members at Iowa State university and EnSoft, DARPA contracts FA8750- 12-2-0126 & FA8750-15-2-0080

Let us first discuss: Is there really a big need to abstract software? What is it?

Have you ever abstracted software?

What are some of the abstractions you have learnt?

One abstraction that all of us have learnt?

- Extremely fundamental.
- Super easy for us to use.
- Super hard for us to invent on our own.

What abstraction comes to your mind?

Ralph is a forest ranger. From each of the four windows of his lookout station, Ralph can see three campsites. The north window shows campsites A, B, C; the east window shows campsites C, E, H; the south window shows campsites F, G, H; the west window shows campsites A, D, F. One day Ralph counted five campers through each of the four windows. There were 14 campers altogether and at least one camper at each campsite. What are the all possible arrangements of the campers in the campsites, i.e. how many campers at each campsite?


The *Algebra* abstraction

Problem 1: John is 3 years older than Jill. Together their ages add up to their mother's age. Their mother is 45 years old. How old are John and Jill?

Problem 2: A water tank holds 3 gallons of water more than another water tank. Together the two water tanks can hold 45 gallons of water. What is the capacity of each water tank?

Are these different problems?

Without knowledge of algebra, the solution will be by trial and error



$$X - Y = 3$$
$$X + Y = 45$$

Algebra is an abstraction!

The algebra abstraction enables computation to solve problems.

What abstraction comes to your mind?

Ralph is a forest ranger. From each of the four windows of his lookout station, Ralph can see three campsites. The north window shows campsites A, B, C; the east window shows campsites C, E, H; the south window shows campsites F, G, H; the west window shows campsites A, D, F. One day Ralph counted five campers through each of the four windows. There were 14 campers altogether and at least one camper at each campsite. What are the all possible arrangements of the campers in the campsites, i.e. how many campers at each campsite?



$$A+B+C = 5$$

$$C+E+H = 5$$

$$F+G+H = 5$$

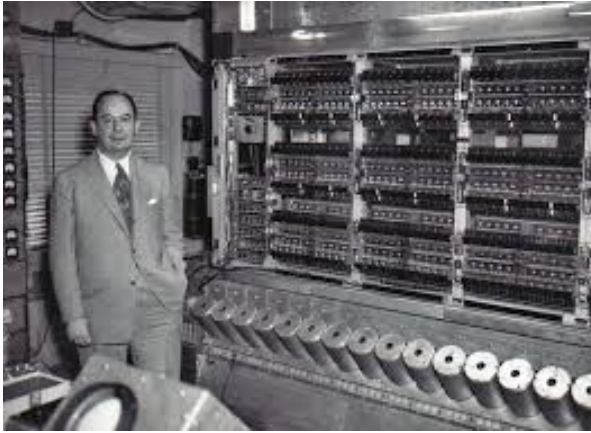
$$A+D+F = 5$$

$$A+B+C+D+E+F+G+H = 14$$

Have you encountered a problem where you could think of multiple abstractions, however, among the many abstractions there was just one *elegant abstraction* to solve the problem?

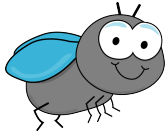
elegant abstraction = simple method to solve the problem + makes us wiser

A story of two abstractions



A car travels from **B** to **A** and fly goes from **A** to **B**. The car as soon as it meets the fly turns back, comes back to **B**, and start again towards **A**. The car does this repeatedly until the fly reaches **B**. What is the total distance travelled by the car? The car and the fly speeds are respectively 20 and 5 miles per hour. The distance AB is 10 miles.

5 miles/hour



distance = 10 miles



A *Abstraction:* D_i : the distance travelled by the car for the i -th meeting
Total distance = sum of the infinite series $D_1 + D_2 + \dots$

B

A loop iteration count problem

```
for i = 1 to N
  for j = 1 to i
    count = count + 1;
```

How many times will the innermost loop iterate?

<i>i</i>	1	2		3						
<i>j</i>	1	1	2	1	2	3				

The answer is $1 + 2 + 3 + \dots + (N-2) + (N-1) + N$

The formula $\frac{1}{2} N(N+1)$ works for counting the number of iterations of a 2-level loop.

An abstraction that would make us wiser?

Consider a k -level nested loop. For example, a loop with $k = 3$

```
for  $i = 1$  to  $N$   
  for  $j = 1$  to  $i$   
    for  $k = 1$  to  $j$   
       $count = count + 1$ ;
```

How many times will the innermost loop iterate?

When did you abstract software – yesterday, last week, last month, last year, or some years ago?

What was the need in your case (to abstract software)?

Was it easy for you (to abstract software)?

What software have you abstracted?

What software do you use? How large are they?

Have you abstracted software X? Who do you think abstracts X?

X = Software you use.

Is it important to abstract software X? Why?

X = Software you use.

Is it easy or hard to abstract software X? Why?

X = Software you use.

Have you used tools to abstract software? What do those tools do? Did the tools help you a lot? How?

What tools are used to abstract software X?

X = Software you use.

- Can we categorize the important needs (to abstract software)?
- What type of tools are needed (to abstract software)?
- What would make the tools useful?

Can we name one important need that all of us encounter (to abstract software)?

What tools are available today to address that need?

Is there a need for a new type of tool? What is it?