Automation: Methodology and Technology

Panel Discussion 3

AUGPath

CUG CS

December 1, 2022

Contents

1 Background

2 Principles of Automation

3 Application

Compute

- “computer” ↔ “person who computes”

Compute

- “computer” ↔ “person who computes”
- complicated → methodology

Contents

1 Background

2 Principles of Automation

- Abstraction
- Algorithms

3 Application

(Part of) Principles

- Abstraction
- Algorithms

Why abstraction?

To make problems clear

(I) To make problems clear.

Why abstraction?

To make problems clear

(I) To make problems clear.

Example

A **farmer(P)** wants to cross a river and take with him a **wolf(W)**, a **goat(G)**, and a **cabbage(C)**.

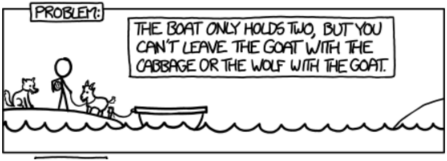
Why abstraction?

To make problems clear

(I) To make problems clear.

Example

A **farmer(P)** wants to cross a river and take with him a **wolf(W)**, a **goat(G)**, and a **cabbage(C)**.

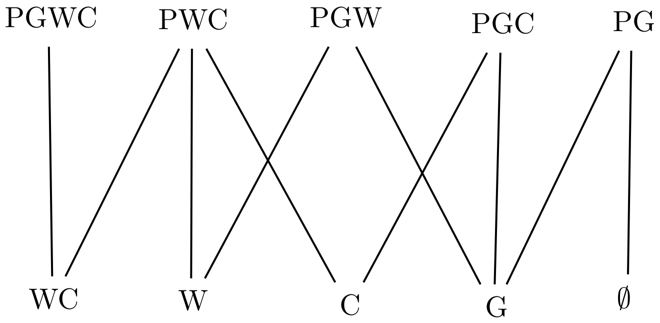

PROBLEM:
THE BOAT ONLY HOLDS TWO, BUT YOU CAN'T LEAVE THE GOAT WITH THE CABBAGE OR THE WOLF WITH THE GOAT.

How can the farmer bring the wolf, the goat, and the cabbage across the river?

Why abstraction?

To make problems clear

Vertex = state of original shore.
Edge = Possible transition that can be made
Farmer=P, Wolf=W, goat=G, Cabbage=C.
That is, find the shortest path of the given graph.



And it's easy to solve now!

Why Abstraction?

Easy to Maintain

(II) Easy to Maintain
Black-box abstraction: What is it about.

Example

We have AND gates and NOT gates and so on...

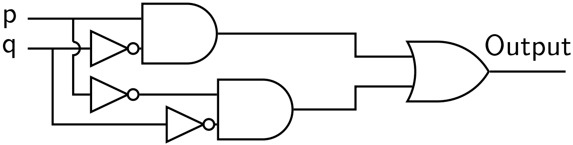
Why Abstraction?

Easy to Maintain

(II) Easy to Maintain
Black-box abstraction: What is it about.

Example

We have AND gates and NOT gates and so on...
We have some wires to construct a functional logic gate.



Why Abstraction?

Easy to Maintain

Black-box abstraction: More precisely...

- Basic Elements: something that are pretty basic.(like sets in Maths)
- Means of Combination: may construct something rather complicated(composition of functions, etc.)
- Means of Abstraction: investigate how can we abstract things(like fixed patterns in math problems)
- Capturing Common Patterns: find how we make the abstractions (like reflection and summarizing after solving a problems)

The black-box abstraction uses the idea of abstraction itself!

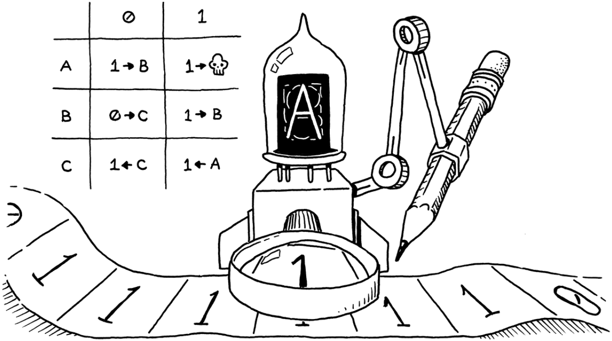
Why Abstraction?

Friendly to represent Data

(III) Friendly to represent Data

machine to automate things → computers

- state of **automation machine** is limited
- a “translation” from real-world problems to **automation machine**



	0	1
A	1→B	1→☠
B	0→C	1→B
C	1←C	1←A

How to Abstract?

Algorithms' Help

Make "abstractions" dynamic

Find the page of word s in *Oxford Advanced Learners' Dictionary*

Make “abstractions” dynamic

Find the page of word s in *Oxford Advanced Learners’ Dictionary*

Example (Find the page of word s(assuming no spelling mistakes) in OALD)

Algorithm 1.

```
for *word* in *dictionary*,
    if *word* is equal to $s$,
        return the page of $s$
```

Make “abstractions” dynamic

Find the page of word s in *Oxford Advanced Learners’ Dictionary*

Example (Find the page of word s(assuming no spelling mistakes) in OALD)

Algorithm 1.

```
for word in dictionary,
    if word is equal to $s$,
        return the page of $s$
```

Algorithm 2.

```
find word in (start page, end page)
Open to the $middle(\lfloor(startpage + endpage)/2\rfloor)$
    Look at page
    If the word is on the page, return the page number.
    If the word is earlier in the dictionary, find word in(start page, middle)
    If the word is later in the dictionary, find word in (middle+1, end page)
```

That’s it

But make sure that you have proved…

- your algorithm is correct
- your algorithm is (somehow) optimized

Contents

1. Background

2. Principles of Automation

3. Application

ETC System

Efforts made in the field of abstraction and algorithms

- huge **database** system → Abstraction(II) (III)
  - data racing, concurrency problems → (Algorithm)
  - efficiency →(Algorithm)
- signals received by receiver → Abstraction(I), (III)

Automation Production in Factory

Efforts made in the field of abstraction and algorithms

- simulation process → Abstraction(I, II)
- stabilize the body of the robots → Algorithm

Dish washing

Efforts made in the field of abstraction and algorithms

- the “washing process” → Algorithm
- the construct of the machine → Abstraction(II)

Verify Mathematical Proofs

Efforts made in the field of abstraction and algorithms

- rules about logic → Abstraction(I)
  - ▶ if p is a prop. , $\neg\neg p \leftrightarrow p$
  - ▶ $A \wedge (B \wedge C) = (A \wedge B) \wedge C$
  - ▶ ...

Lean Theorem Prover

`http://leanprover-community.github.io/lean-web-editor`

Verify Mathematical Proofs

Example

```
variables A B C D : Prop
variable h1 : A -> B -> C
variable h2 : D -> A
variable h3 : D
variable h4 : B
#check h2 h3
#check h1 (h2 h3)
#check h1 (h2 h3) h4
```

More stuff: https://leanprover.github.io/theorem_proving_in_lean4/title_page.html

Data Providers on Web

gets information by...

- web crawler
- government files
- official dataset

before automatically process these data.

Summary and References

[1] *Problem Solving 2020*, Nanjing University.
[2] *Minecraft Logic Gates*, FandomWiki.
[3] *Structure and Interpretation of Computer Programs, 1986*, MIT.
[4] *CS50x 2022*, Harvard University.
[5] *Logic and Proof* by Jeremy Avigad, Robert Y. Lewis, and Floris van Doorn.

Thanks!