Problem #7: *Smart water*
International Physics Tournament

Team Poland/Kraków (Jagiellonian University)

14/3/2023

Krzysztof Cofałka, **Bartosz Grygielski**, Karol Orkisz, Ksymena Poradzisz, **Adam Sowiński**,
Franciszek Ungeheuer, Mateusz Winiarski, Anastasiya Yushchuk, Wiktor Zantowicz
Captain: Mateusz Winiarski
Supervisors: dr hab. Tomasz Kawalec, mgr inż. Anna Władyszewska

UNIWERSYTET JAGIELLOŃSKI
W KRAKOWIE

INTERNATIONAL
PHYSICISTS'
TOURNAMENT

Presentation plan

Problem statement

If you let water flow in a maze you end up with a physical maze-solving algorithm.
Many other algorithms and computations can be done with physical systems.
Create a water-based computer (without electronics) that solves an NP-hard problem.

Algorithm complexility

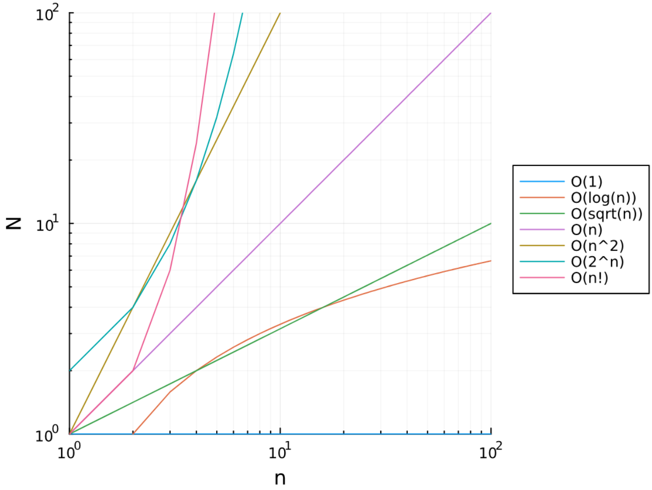


Figure: Algorithm complexity graph

P vs NP

- P: solvable in polynomial time by a deterministic Turing machine
- NP: solvable in polynomial time by a nondeterministic Turing machine

Intuition: quickly solvable vs quickly checkable

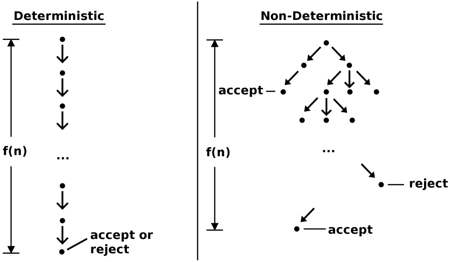


Figure: Deterministic vs non-deterministic Turing machine (source: Wikimedia Commons)

SAT problem

SAT: A Practical Example

- Consider the following constraints:
  - John can only meet either on Monday, Wednesday or Thursday
  - Catherine cannot meet on Wednesday
  - Anne cannot meet on Friday
  - Peter cannot meet neither on Tuesday nor on Thursday
  - QUESTION: When can the meeting take place?
- Encode then into the following Boolean formula:
  $(Mon \vee Wed \vee Thu) \wedge (\neg Wed) \wedge (\neg Fri) \wedge (\neg Tue \wedge \neg Thu)$
  - The meeting *must* take place on *Monday*

Figure: Example of SAT problem (source: Larrosa et al. Satisfiability: Algorithms, Applications and Extensions)

Applications of SAT

- There is at least one number in each entry:
  $\bigwedge_{x=1}^{9} \bigwedge_{y=1}^{9} \bigvee_{z=1}^{9} s_{xyz}$
- Each number appears at most once in each row:
  $\bigwedge_{y=1}^{9} \bigwedge_{z=1}^{9} \bigwedge_{x=1}^{8} \bigwedge_{i=x+1}^{9} (\neg s_{xyz} \vee \neg s_{iyz})$
- Each number appears at most once in each column:
  $\bigwedge_{x=1}^{9} \bigwedge_{z=1}^{9} \bigwedge_{y=1}^{8} \bigwedge_{i=y+1}^{9} (\neg s_{xyz} \vee \neg s_{xiz})$
- Each number appears at most once in each 3x3 sub-grid:
  $\bigwedge_{z=1}^{9} \bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{x=1}^{3} \bigwedge_{y=1}^{3} \bigwedge_{k=y+1}^{3} (\neg s_{(3i+x)(3j+y)z} \vee \neg s_{(3i+x)(3j+k)z})$
  $\bigwedge_{z=1}^{9} \bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{x=1}^{3} \bigwedge_{y=1}^{3} \bigwedge_{k=x+1}^{3} \bigwedge_{l=1}^{3} (\neg s_{(3i+x)(3j+y)z} \vee \neg s_{(3i+k)(3j+l)z})$.

Figure: SAT minimal encoding for sudoku (source: Lynce, I., & Ouaknine, J. (2006, January). Sudoku as a SAT Problem. In AI&M.

3SAT problem

$$(A \vee B \vee C) \wedge (\neg A \vee \neg B \vee D) \wedge (B \vee \neg C \vee \neg D)$$

Sketch

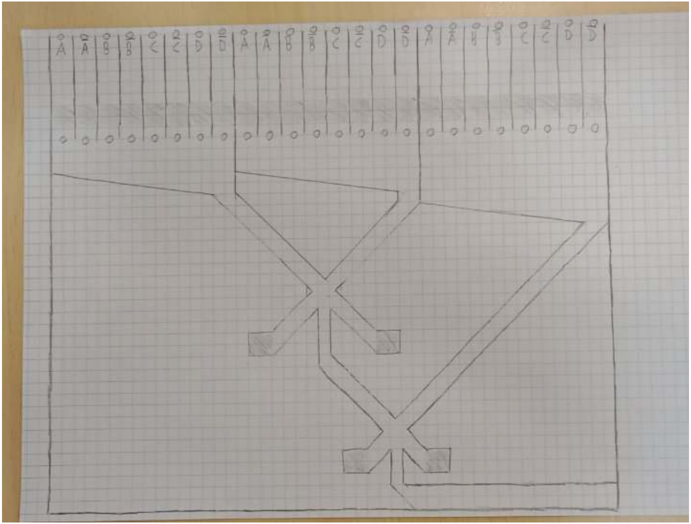


Figure: First sketch of KRAK3SAT water computer module

Principle of working

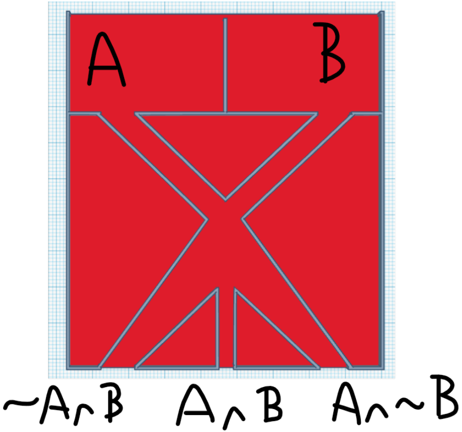
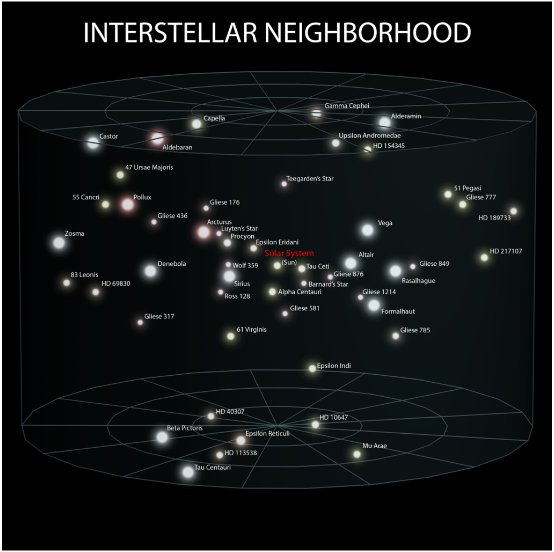


Figure: Model of AND gate used by water computer

What can be done (theoretically)



Conclusions

- We constructed a water computer
- It works