Satisfiability

A Lecture in CE Freshman Seminar Series:
Ten Puzzling Problems in Computer Engineering
About This Presentation

This presentation belongs to the lecture series entitled “Ten Puzzling Problems in Computer Engineering,” devised for a ten-week, one-unit, freshman seminar course by Behrooz Parhami, Professor of Computer Engineering at University of California, Santa Barbara. The material can be used freely in teaching and other educational settings. Unauthorized uses, including any use for financial gain, are prohibited. © Behrooz Parhami

<table>
<thead>
<tr>
<th>Edition</th>
<th>Released</th>
<th>Revised</th>
<th>Revised</th>
<th>Revised</th>
<th>Revised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apr. 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Making Change: Ground Rules

For the sake of the puzzles in this presentation:
Coin means one of the pieces 1¢, 5¢, 10¢, 25¢ (no 50¢ or $1 coin)
Bill means one of the denominations $1, $5, $10, $20, $50, $100

Warm-up puzzle: What is the largest sum of coins that you can have in your pocket without being able to make exact change for $1?Answer: $1.19
Making Change: Mathematical Formulation

**Puzzle:** Largest sum of coins not containing exact change for $1.

**Formulating a solution:** Maximize the sum \( M = 25Q + 10D + 5N + P \) subject to \( 25q + 10d + 5n + p \neq 100 \) for any \( q \leq Q, d \leq D, n \leq N, p \leq P \)

**Challenge:** You held some bills, all $50 or smaller denominations, and some coins, when a friend asked you if you could change a $100 bill for her. You discovered that you could not make exact change. What is the maximum sum of money that you could have had?

Answer: $140.19
The Many Ways of Making Change

In how many distinct ways can you make change for $1?

Count the number of integer solutions to the equation $25Q + 10D + 5N + P = 100$

Because $P$ must be a multiple of 5, we take $P = 5P'$ and rewrite the equation as $5Q + 2D + N + P' = 20$

Diophantine equation

---

**Q:**
- 4
  - 4,0,0,0

**D:**
- 2
  - 3,2,1,0
  - 3,2,0,5

**N + P':**
- 1
  - 2D + N + P' = 5
- 3
  - 2D + N + P' = 10
- 5
  - 2D + N + P' = 15
- 20
  - 2D + N + P' = 20

---

**N:**
- 0
  - 0,10,0,0
  - N + P' = 2
  - N + P' = 20

---

**P':**
- 0
  - 10
  - 9
  - ... 0
Can you make change for $1 using exactly 22 coins?

Is there a solution to \(25Q + 10D + 5N + P = 100\) that satisfies the constraint \(Q + D + N + P = 22\)?

\[P = 22 - Q - D - N\]
\[24Q + 9D + 4N = 78\]

- \(Q\): 3, 2, 1, 0
  - 9D + 4N = 6
    - No solution
  - 9D + 4N = 30
  - 9D + 4N = 54
  - 9D + 4N = 78

- \(D\): 2, 0, 6, 4, 2, 0
  - 4N = 30
    - No solution
  - 4N = 6
    - No solution
  - 4N = 42
    - No solution
  - 4N = 78
    - No solution

Quarters, Dimes, Nickels, Pennies
What Is (Boolean) Satisfiability?

We converted the puzzle:

Can you make change for $1 using exactly 22 coins?

To the equivalent problem:

Is the equation \(24Q + 9D + 4N = 78\) satisfiable for \(Q, D,\) and \(N\) nonnegative integers?

Inputs and outputs of logic circuits are in \(\{0, 1\}\)

AND gate: inputs \(x, y, z\); output \(xyz\) (product of inputs)

OR gate: inputs \(x, y, z\); output \(1 - (1 - x)(1 - y)(1 - z)\)

NOT gate: input \(x\); output \(1 - x\)

Satisfied (=1) for:

- \(b = 1, a = 1;\) or
- \(b = 0, c = d = 0\)
Why Is Satisfiability Difficult?

As was the case for our money-changing puzzles, the tree of options quickly expands in size, thus leading to a great deal of work for extracting a solution from the vast array of possibilities.

The problem is much easier going the other way: Given the circuit input values, what is its output?
A Restricted Form of Satisfiability

The logic circuit consists of a number of OR gates, each with up to 3 inputs, all feeding an AND gate that produces the output $y$.

Each circuit input is a variable $x_i$ or $1 - x_i$.

Even this restricted version, known as 3SAT, isn’t fundamentally simpler than the general version.

Example: Does the expression below evaluate to 1 for some set of values in $\{0, 1\}$ assigned to $x_1$-$x_9$?

\[
1 - (1 - x_7)x_8(1 - x_9) \\
1 - x_9(1 - x_3)x_8 \\
1 - (1 - x_5)x_7x_8 \\
1 - (1 - x_6)x_2x_5 \\
1 - (1 - x_8)(1 - x_1)(1 - x_2) \\
1 - x_4(1 - x_9)(1 - x_6)
\]

Answer: Yes, e.g., for $x_1 = x_3 = x_6 = x_9 = 1$ and $x_8 = 0$.
Why Is Satisfiability Useful?

Suppose that the circuit illuminates an emergency light when \( f = 1 \). You operate the circuit for weeks and the light never goes on. Is this an indication that everything is okay, or a sign that the circuit’s output has been disconnected from the emergency light?

The satisfiability result that we obtained tells you that you can test the circuit for \( f \) stuck-at-0 by applying any of the inputs \((a \ b \ c \ d) = 1 \ 1 \ 1 \ 0 \ 0\) or \((a \ b \ c \ d) = 0 \ 0 \ 0 \ 0\), for a total of 6 possible test inputs.

**Challenge:** If the light goes on, how can you test the circuit to determine whether this is due to \( f \) being stuck-at-1 rather than a real emergency?
Improving Testability via Testpoint Insertion

Inputs → Output

Inputs → Outputs
Equivalence of Digital Circuits

Question: Are C1 and C2 equivalent (always produce the same output)?

Circuits C1 and C2 are equivalent iff C3 is not satisfiable

Output of XOR gate is 1 iff its two inputs are unequal

\[ y = x + y - 2xy \]
The Satisfiability Game

Pick one of the numbers shown in the table (say $x$)

Color all boxes containing $x$ green
and all boxes containing $-x$ red

You win if every row has a green box
You lose if some row has three red boxes

(a) Pick $-8$
(b) Pick $6$
(c) Pick $3$

It is clear that we can win by choosing one number from each of the sets $\{-4, 9\}$ and $\{1, 2\}$

Play the satisfiability game on-line at the following website. The game has five difficulty levels.

The Satisfiability Game Is the Same as SAT3

Each OR gate corresponds to a row in the table.

Having a green box in a row is the same as the associated OR gate having an input 1.

Having a red box in a row is the same as the associated OR gate having an input 0.

We win if all OR gates have outputs 1; We lose if some OR gate has output 0 (row of 3 red boxes).
A Possible “Strategy” for SAT3

At each step, pick a number that produces the most green boxes and as few red boxes as possible

Consider the example on the right, where most of the entries are not shown to avoid clutter

(a) Pick 9; Satisfies 3 rows
(b) Pick –7: Satisfies 3 rows
(c) Pick 5: Satisfies 3 rows

This seemingly reasonable strategy does now work because the 3 choices lead to an unsatisfiable row

For any alleged strategy, one can build a specific counterexample that would defeat that strategy

Exercise: Fill in the rest of this SAT game in a way that makes it difficult for a player following the “strategy” to succeed
Another Application of Satisfiability

**Map labeling:** We have a map with \( n \) cities which must be labeled. Let’s say the label for a city can go above \( (X = 1) \) or below \( (X = 0) \) it. When cities are close to each other, we have constraints such as \( B = 1 \) and \( C = 0 \) are not allowed simultaneously, which can be written as \( B(1 − C) = 0 \) or equivalently \( 1 − B(1 − C) = 1 \).

**Problem:** Choose the Boolean variables A-G such that the following equation is satisfied (an instance of 2SAT):

\[
[1 − B(1 − C)] [1 − D(1 − B)] [1 − E(1 − C)] [1 − G(1 − F)] = 1
\]

Satisfied, e.g., for
\[
A = D = E = G = 0 \\
B = C = F = 1
\]