

Project 2 - McMillon

April 14, 2024

08/30/23

09/06/23

0.1 Gallager A/B

1. Only flip Variable Node (VN) if all neighboring Check Nodes (CN) are unsatisfied.
2. Flip when at least half of the neighboring check nodes are unsatisfied.
3. Flip when any CN neighbor is unsatisfied.

9/13/23

Names for writing things!

1. **Gretchen: We have colors!**
2. **Emily: Use them!**
3. **Andrea: This way we can tell who has written what in the document.**
4. **Collin: The colors are ok?**
5. **Julian: Latex has a limited number of predefined colors.**
6. **Liz: So we do our best.**
7. **Name: With what we have...**
8. **Renaya: Hopefully this works well!**

Emily: Agenda:

1. Ask about reading.
2. Define absorbing sets rigorously, draw some, explain why decoding fails.
3. Activity: work on if the threshold is different, i.e., if the number of unsatisfied check nodes required to flip is at least 1. Try to explore/make conjectures using graphs with 3-4 variable nodes.

9/27/23

In our standard definition of absorbing set, if a check node has degree $\deg(v) = u + s$ where s is the number of satisfied check nodes and u is the number of unsatisfied check nodes, our decoder rule is that we change the value of v if $u \geq s$. What happens if we change the decoder rule so that:

- We change the value of v if $u \geq s + 1$.
- We change the value of v if $u \geq s - 1$.

The answer should be a characterization of types of graphs, similar to the characterization of absorbing sets, where the number of odd degree neighbors of any variable node was strictly less than the number of even degree variable nodes. It might be good to start with some conventional absorbing sets and alter them.

1 9/28/23

10/4/2023

When is a standard absorbing set still an absorbing set? What if our update rule is $u \geq s + 1$ or $u \geq s - 1$?

10/11/2023

Prove that absorbing sets with the $u \geq s$ update rule are still absorbing sets with the $u \geq s + 1$ update rule.

Definition 1.1. A $(a | b)$ absorbing set is a set of variable nodes that, under the $a \geq b$ update rule, the set of variable nodes is not eventually correct.

Theorem 1.2. All $(u | s)$ -absorbing sets are $(u | s + 1)$ -absorbing sets.

Proof. Let V denote the set of variable nodes. Assume $(u | s)$ is an arbitrary absorbing set, meaning for all $v \in V$, $D_e(v) > D_o(v)$ holds. Initializing all variable nodes with 1, we have $D_e(v) + 1 > D_o(v)$ for all $v \in V$. Therefore, $u \geq s + 1$ is never satisfied, implying that $(u | s)$ is also a $(u | s + 1)$ -absorbing set. \square

Theorem 1.3. All $(u | s)$ -absorbing sets are $(u | s + k)$ -absorbing sets.

Conjecture 1.4. Not all $(u | s)$ -absorbing sets are $(u | s - 1)$ -absorbing sets.

Gretchen's challenge: Can you come up with things that are $(u | s + 1)$ -absorbing sets but not $(u | s)$ -absorbing sets?

If $(u | s)$ implies $(u | s + k)$, does that mean that $(u | s - k)$ implies $(u | s)$?

Corollary 1.5. If $(u | s)$ is an absorbing set then $(u | s + k)$ is an absorbing set where $k \in \mathbb{N}_0$

Proof. Assume that $(u | s)$ is an absorbing set and that there exists some $k \in \mathbb{N}$ such that $(u | s + k)$ is not an absorbing set. Then, by the well-ordering principle, there exists the smallest $k \in \mathbb{N}$ such that $(u | s + k)$ is not an absorbing set. As $(u | s + k)$ is the smallest k that is not an absorbing set, then $(u | s + k - 1)$ is an absorbing set. By Theorem 1.2, $(u | s + k)$ is also an absorbing set, a contradiction. Therefore, $(u | s + k)$ is an absorbing set. \square

Proof. Assume that $(u | s)$ is an absorbing set. Then set $k = 0$ as therefore $(u | s + 0)$ is also an absorbing set. Now assume that $\exists k \in \mathbb{N}$ s.t. $(u | s + k)$ is an absorbing set. Then by Theorem 1.2 $(u | s + k + 1)$ is also an absorbing set. \square

10/25/2023

Theorem 1.6. *If every $v \in S$ is such that v has exactly t more even degree than odd degree neighbors in G_S , then S is not an $(u \mid s - t)$ absorbing set. It is a $(u \mid s - k)$ absorbing set for all $k < t$.*

Proof. Assuming that for all $v \in S$, the equality $D_e(v) = D_o(v) + t$ holds, where $t \in \mathbb{N}_0$. If every variable node is initialized with 1, then $D_e(v) = s$ and $D_o(v) = u$. Under the update rule $u \geq s - t$, this would not be an absorbing set, as $D_o(v) = D_e(v) - t$ would cause every variable node to switch on the first iteration of the algorithm.

Under the update rule $u \geq s - k$ where $k < t$, this would be an absorbing set as $u \geq u + t - k$ would never be satisfied. \square

Theorem 1.7. *If every $v \in S$ is such that v has at least t more even degree than odd degree neighbors in G_S , then S is a $(u \mid s - t + 1)$ absorbing set.*

Proof. Assuming that for all $v \in S$, the inequality $D_e(v) \geq D_o(v) + t$ holds, where $t \in \mathbb{N}_0$. If every variable node is initialized with 1, then $D_e(v) = s$ and $D_o(v) = u$. Under the update rule $u \geq s - t + 1$, this would not be an absorbing set, as $s + 1 \leq u + t \leq s$ is never satisfied. \square

Question: How can we identify these new absorbing sets under new update rules by looking at matrices?

(Poorly phrased) Question: What is the least amount of things we can do to make something not an absorbing set any more?

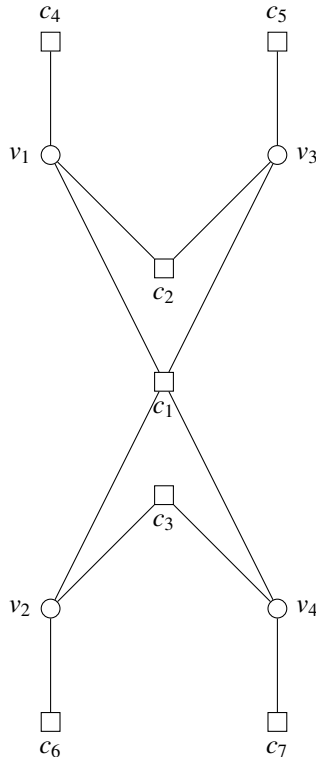
11/8/2023

The number of odd degree check nodes for a variable node is

$$D_o(v) = |\{c \in N(v) : |N(c)| \equiv 1 \pmod{2}\}|$$

The number of even degree check nodes for a variable node is

$$D_e(v) = |\{c \in N(v) : |N(c)| \equiv 0 \pmod{2}\}|$$



Corresponding matrix to the above graph:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question: What relationship does a graph having a nonzero solution (i.e. all check nodes are satisfied with a set of variable node assignments such that not every variable node assignment is zero) to being an absorbing set?

11/15/2023

Corrected definitions:

- A trapping set is a set of variable nodes that does not eventually converge to a solution under a set iterative decoding rule.
- An absorbing set is a set of variable nodes that, assuming all variable nodes have an input of 1, under our iterative decoding rule, no variable node changes its value. (i.e. no 1 flips to a 0)
- In the classical $(u|s)$ setting, the above definition translates to for all $v \in D$, in G_D , $D_o(v) < D_e(v)$.

Prove this: Can we classify absorbing sets in $(u | s + k)$? Prove that this is when $D_o(v) \leq D_e(v) + k - 1$. Yes, and this is a full characterization of absorbing sets under the $(u | s + k)$ update rule.

Proof. Let the update rule be $u \geq s + k$ and assume that for all $v \in S$ the inequality $D_o(v) \leq D_e(v) + k - 1$ is true. Then initializing all variable nodes with 1 then the for all variable nodes $u \leq s + k - 1$ is true. Then changing to a strict inequality we have $u < s + k$ therefore no variable nodes would satisfy the update rule $u \geq s + k$. \square

Fix an update rule. What is the minimal operation needed to break an absorbing set? What if we have to keep the code the same?

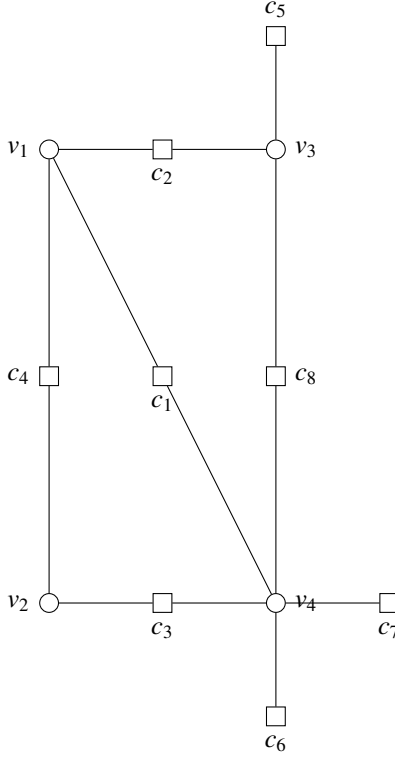
What if we change our update rule to be a fraction?

11/29/2023

If we fix a class of codes, can we talk about breaking absorbing sets in a smart way that doesn't add new ones?

12/6/2023

Collin's additions:



If we are only interested in determining whether something is in the set of words, then we can treat the graph as a Boolean function and use tools for finding equivalent functions. We can have the check nodes each be their own individual Boolean function that will yield zero if and only if the sum of the variable nodes is even. This can be done using the parity function; in this case, as they all have a degree of two, the parity function would simply be XOR. For example, looking at the check node c_2 , we could create the Boolean function $c_2(a, b) = a\bar{b} + \bar{a}b$, where multiplication represents logical 'and,' addition is logical 'or,' and the overline denotes logical negation. Now, $c_2(1, 0) = 1\bar{0} + \bar{1}0 = 1 \cdot 1 + 0 = 1$. So, if the output of the function is 1, then we know the number of 1's in the input was odd. If we repeat this process for the entire graph and add each individual check node in the parity function form, we get:

$$g(v_1, v_2, v_3, v_4) = v_1\bar{v}_3 + \bar{v}_1v_3 + v_3 + v_3\bar{v}_4 + \bar{v}_3v_4 + v_4 + v_4 + v_4\bar{v}_2 + \bar{v}_4v_2 + v_1\bar{v}_2 + \bar{v}_1v_2 + v_1\bar{v}_4 + \bar{v}_1v_4$$

After simplifying, we find that this function is equal to:

$$g(v_1, v_2, v_3, v_4) = v_1 + v_2 + v_3 + v_4$$

A word is in the alphabet A if and only if, after plugging it into the function, it is equal to 0. That is, $a \in A \iff g(a) = 0$. The tool I used to reduce the complexity of the function is called a Karnaugh map. This has no real use as you lose all information that matters to decoding the word as you do not have any information on satisfied and unsatisfied check nodes.