Beam Algorithm - Software Version (v0.0.5)

1. Initialize a groupoid G of minimum size 3x3. Initialize a beam of width w and the set of male terms M equal the set of term variables $\overrightarrow{x}$. Initialize T, the target array of length g^k where g is the size of the groupoid G and k is the number of term variables in $\overrightarrow{x}$.

2. At beam level 0, initialize a set F containing w empty female terms $f_0(\Diamond)$. Let variable H be the current level of a female term in the beam, initialized to 0.

3. At beam level 0 mate each female term $f_0(\Diamond)$ with each male term $m(\overrightarrow{x})$ in M and check if the resulting offspring $f_0(m(\overrightarrow{x}))$ is a solution to the target array T. If $f_0(m(\overrightarrow{x}))$ is a solution to T, then return $f_0(m(\overrightarrow{x}))$.

4. Start a process $P_{f_H(\overrightarrow{x},\Diamond)}$ for each female term $f_H(\overrightarrow{x},\Diamond)$ in F for a total of w processes. Let each process $P_{f_H(\overrightarrow{x},\Diamond)}$ search for a female term at level $H+1$ called $f_{H+1}(\overrightarrow{x},\Diamond)$ that is valid with respect to the validity array of $f_H(\overrightarrow{x},\Diamond)$.

5. Let variable C be the number of child terms of some term $f_H(\overrightarrow{x},\Diamond)$ that are required to promote a proccess $P_{f_H(\overrightarrow{x},\Diamond)}$ to level $H+1$ before a higher beam level is full. A beam level is considered full when there are w valid female terms at that level.

START LOOPING CONTINUOUSLY

6. Let $f_{H+1}(\overrightarrow{x},\Diamond)$ be a valid female term returned by a process $P_{f_H(\overrightarrow{x},\Diamond)}$. Assume $f_{H+1}(\overrightarrow{x},\Diamond)$ was found using female term candidates chosen at random inside of a process $P_{f_H(\overrightarrow{x},\Diamond)}$ using the Gamblers Ruin Algorithm (GRA). Add $f_{H+1}(\overrightarrow{x},\Diamond)$ to beam level $H+1$.

7. Let $f_H(\overrightarrow{x},\Diamond)$ be the parent term at level H of the valid female term $f_{H+1}(\overrightarrow{x},\Diamond)$ found in step 6.

8. Mate the valid female term $f_{H+1}(\overrightarrow{x},\Diamond)$ (from 6) with each male term $m(\overrightarrow{x})$ in M and check if the resulting offspring $f_{H+1}(\overrightarrow{x},m(\overrightarrow{x}))$ has a term operation that is a solution to the validity array of $f_H(\overrightarrow{x},\Diamond)$. If $f_{H+1}(\overrightarrow{x},m(\overrightarrow{x}))$ is a solution to the validity array of $f_H(\overrightarrow{x},\Diamond)$, then break from the loop and proceed to step 12.

9. If $f_H(\overrightarrow{x},\Diamond)$ has produced C children at H+1:

   Assign process $P_{f_H(\overrightarrow{x},\Diamond)}$ to the child term $f_{H+1}(\overrightarrow{x},\Diamond)$ at level H+1 and have it search for some term $f_{H+2}(\overrightarrow{x},\Diamond)$ that is valid with respect to validity array of $f_{H+1}(\overrightarrow{x},\Diamond)$.

Next let PL be the ordered set of processes running at a level below H+1 and sorted ascending by proccess level and number of produced child terms. For each process $P_{f_{LH}(\overrightarrow{x},\Diamond)}$ in PL, kill $P_{f_{LH}(\overrightarrow{x},\Diamond)}$ and assign $P_{f_{LH}(\overrightarrow{x},\Diamond)}$ to the next child of $f_H(\overrightarrow{x},\Diamond)$ that doesn't already have a running process $P_{f_{H+1}(\overrightarrow{x},\Diamond)}$ associated with it. Have process $P_{f_{LH}(\overrightarrow{x},\Diamond)}$ search for for some term $f_{H+2}(\overrightarrow{x},\Diamond)$ that is valid with respect to the validity array of that child of $f_H(\overrightarrow{x},\Diamond)$ that it was assigned to. Once all child terms of $f_H(\overrightarrow{x},\Diamond)$ have an assigned running process or there are no more processes running below level H+1, return to step 6.

10. If the beam is full at a level above the level of the lowest running process:

    Let PL be the set of processes running at a level below the highest full level. Kill all processes $P_{f_{LH}(\overrightarrow{x},\Diamond)}$ in PL, assign them to terms at the highest full level, and have each of them search for a new female term that is valid with respect to the array of the term that they were respectively assigned to. Return to step 6.

11. If conditions 9 and 10 both were not satisfied, then rerun process $P_{f_H(\overrightarrow{x},\Diamond)}$ for the parent $f_H(\overrightarrow{x},\Diamond)$ of the valid female term $f_{H+1}(\overrightarrow{x},\Diamond)$. Essentially don't reassign any processes to a higher level and continue searching for another female term$f_{H+1}(\overrightarrow{x},\Diamond)$ that is valid with respect to $f_H(\overrightarrow{x},\Diamond)$. Return to step 6.

CONTINUE LOOPING CONTINUOUSLY

---

12. Some solution term $f_H(\overrightarrow{x},m(\overrightarrow{x}))$ was found at step 8. Recursively mate $f(\overrightarrow{x},m(\overrightarrow{x}))$ with each of parent term at $f_{H-1}(\overrightarrow{x},\Diamond)$, until reaching the term that has no parent. The result is a term which has an array that is a solution to the target array T.