Beam Algorithm - Software Version (v0.0.6)

1. Initialize algorithm variables.
   (a) Initialize G, a groupoid of minimum size 3x3.
   (b) Initialize w, the beam width of minimum size 1.
   (c) Initialize H, the current level of a female term in the beam, to 0.
   (d) Initizlize M, the set of candidate male terms, to the set of term variables $\overrightarrow{x}$.
   (e) Initialize T to 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 $\overrightarrow{x}$.
   (f) Initialize $lrlc$ to an integer value ≥ 0 that determines the number of beam levels that should use the solution term for the left (LA) or right array (RA) of some target A, in order to produce a female term that is valid wrt A.
   (g) Initialize F to a set containing w empty female terms $f_0(\Diamond)$ at beam level 0.
   (h) Initialize $P_{f_H(\overrightarrow{x},\Diamond)}$, a process 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 wrt the validity array of $f_H(\overrightarrow{x},\Diamond)$.
   (i) Initialize pcc, a positive integer that specifies 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.
2. Define subalgorithms.
   (a) Define *Valid Female Term Generation Method 1* as a method for finding valid female terms by using the GRA to produce a term $u(\overrightarrow{x})$ and then check both $u(\overrightarrow{x})\Diamond$ and $\Diamond u(\overrightarrow{x})$ for validity with respect to the validity array of $f_H(\overrightarrow{x},\Diamond)$.
   (b) Define *Valid Female Term Generation Method 2* as a method for finding valid female terms by randomly choosing L or R. If L, then, for each GRA term $u(\overrightarrow{x})$, check $u(\overrightarrow{x})\Diamond$ for validity wrt to LA where A is the validity array of $f_H(\overrightarrow{x},\Diamond)$. If $u(\overrightarrow{x})\Diamond$ is valid wrt the the validity array of LA, take $u(\overrightarrow{x})\Diamond$ to be the term at $f_{H+1}(\overrightarrow{x},\Diamond)$. If R, then, for each GRA term $u(\overrightarrow{x})$, check $\Diamond u(\overrightarrow{x})$ for validity wrt to RA where A is the validity array of $f_H(\overrightarrow{x},\Diamond)$. If $u(\overrightarrow{x})\Diamond$ is valid wrt the the validity array of RA, take $\Diamond u(\overrightarrow{x})$ to be the term at $f_{H+1}(\overrightarrow{x},\Diamond)$.

(c) Define *Child Promotion Method 1* as a method for reassigning processes when a process has produced *pcc* child terms. 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 wrt validity array of $f_{H+1}(\overrightarrow{x},\Diamond)$.

Next let PL be the ordered set of processes actively 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 wrt the validity array of that child of $f_H(\overrightarrow{x},\Diamond)$ that it was assigned to.

(d) Define *Child Promotion Method 2* as a method for reassigning processes when 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 wrt the array of the term that they were respectively assigned to.

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 $f_0(m(\overrightarrow{x}))$.

START LOOPING CONTINUOUSLY

4. Let $f_{H+1}(\overrightarrow{x},\Diamond)$ be a valid female term returned by a process $P_{f_H(\overrightarrow{x},\Diamond)}$ and add $f_{H+1}(\overrightarrow{x},\Diamond)$ to beam level $H+1$. If $H > lrlc$, then assume $f_{H+1}(\overrightarrow{x},\Diamond)$ was found using subalgorithm (a) *Valid Female Term Generation Method 1* . If $H \leq lrlc$, then assume $f_{H+1}(\overrightarrow{x},\Diamond)$ was found using subalgorithm (b) *Valid Female Term Generation Method 2* .

5. Mate the valid female term $f_{H+1}(\overrightarrow{x},\Diamond)$ (from 4) 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 9.

6. If $f_H(\overrightarrow{x},\Diamond)$ has produced *pcc* children at H+1, then proceed with reassigning processes according to subalogrithm defintion (c) *Child Promotion Method 1.* Return to step 4.

7. If the beam is full at a level above the level of the lowest running process, then proceed with reassigning processes according to sub-

alogrithm defintion (d) *Child Promotion Method 2*. Return to step 4.

8. If conditions 6 and 7 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 wrt $f_H(\overrightarrow{x},\Diamond)$. Return to step 4.

CONTINUE LOOPING CONTINUOUSLY

---

9. Some solution term $f_H(\overrightarrow{x},m(\overrightarrow{x}))$ was found at step 5. 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.