

Constraints

- Constraints: (1) size - at most n patterns;
- (2) length (tree depth) - at most k symbols per pattern;
- (3) completeness - each component must begin with a start symbol and end with the end symbol;
- (4) at most k loops per component

Pseudocode

(\ is the diff operator on sets)

1. Initialize **results_set** as empty
2. Separate elems with start symbol **start-elems** and others **other-elems**
3. FOR EACH **start-elem** IN **start-elems** DO
 1. Initialize a tree with a root of **start-elem**
 2. Assign all siblings on a root to be **other-elems**
 3. Check if *constraints* still hold
 4. If among **root.siblings** there is end symbol
 1. Calculate score
 2. write to output
 5. FOR EACH **root.sibling** IN **root.siblings** DO
 1. Assign (**root.siblings** \ **root.sibling**) to **root.sibling**
 2. Do steps 3.3 3.4 3.5 while step 3 returns **True**.

// Now greedy part of the algorithm

1. Initialize **final_patterns**
2. Sort the **results_set** in descending order for the score
3. UNTIL FOUND N patterns DO
 1. Take elem **one_pattern** from **results_set**
 2. If all patterns in the **one_pattern** are not in any patterns in **final_patterns**
 1. Add **one_pattern** to **final_patterns**
4. return **final_patterns**