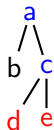


Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- Example: **stem**, **sorted sibling sequence**, **window $w = 3$**

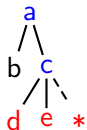


Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

1 if *sibling sequence* < *window* then extend with dummy nodes;

-
- Example: stem, sorted sibling sequence, window $w = 3$

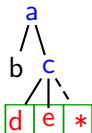


Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- 1 if *sibling sequence* < *window* then extend with dummy nodes;
- 2 **initialize window**: start with leftmost node;

- Example: stem, sorted sibling sequence, window $w = 3$



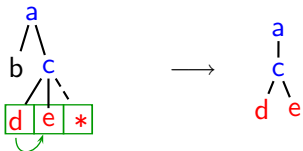
Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- 1 if *sibling sequence* < *window* then extend with dummy nodes;
 - 2 initialize window: start with leftmost node;
 - 3 repeat
 - 4 form bases in window: all q -permutations that contain start node;

 - 7 until *processed all window positions*
-

- Example: stem, sorted sibling sequence, window $w = 3$



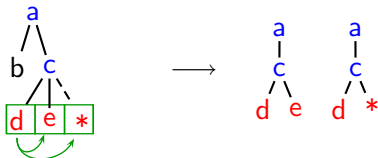
Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- 1 if *sibling sequence* < *window* then extend with dummy nodes;
 - 2 initialize window: start with leftmost node;
 - 3 repeat
 - 4 **form bases** in window: all q -permutations that contain start node;

 - 7 until *processed all window positions*
-

- Example: stem, sorted sibling sequence, window $w = 3$



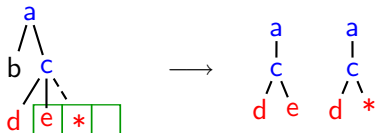
Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- 1 if *sibling sequence* < *window* then extend with dummy nodes;
 - 2 initialize *window*: start with leftmost node;
 - 3 repeat
 - 4 *form bases* in window: all q -permutations that contain start node;
 - 5 *shift window* to the right by one node;

 - 7 until *processed all window positions*
-

- Example: stem, sorted sibling sequence, window $w = 3$

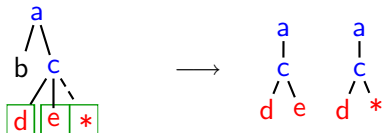


Solution: Windowed pq -Gram Bases

Algorithm 1: Form bases from a sorted sibling sequence

- 1 if *sibling sequence* $<$ *window* then **extend with dummy nodes**;
 - 2 **initialize window**: start with leftmost node;
 - 3 repeat
 - 4 **form bases** in window: all q -permutations that contain start node;
 - 5 **shift window** to the right by one node;
 - 6 if *window extends the right border* then **wrap window**;
 - 7 until *processed all window positions*
-

- Example: **stem**, **sorted sibling sequence**, **window** $w = 3$

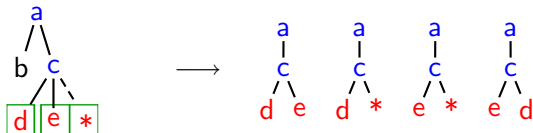


Solution: Windowed pq -Gram Bases

Algorithm 1: Form bases from a sorted sibling sequence

- 1 if *sibling sequence* $<$ *window* then **extend with dummy nodes**;
 - 2 **initialize window**: start with leftmost node;
 - 3 repeat
 - 4 **form bases** in window: all q -permutations that contain start node;
 - 5 **shift window** to the right by one node;
 - 6 if *window extends the right border* then **wrap window**;
 - 7 until *processed all window positions*
-

- Example: **stem**, **sorted sibling sequence**, **window** $w = 3$

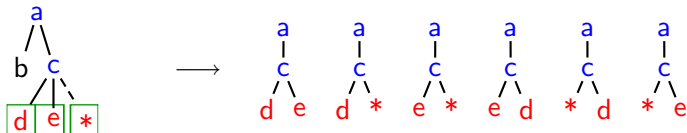


Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- 1 if *sibling sequence* $<$ *window* then **extend with dummy nodes**;
- 2 **initialize window**: start with leftmost node;
- 3 repeat
- 4 **form bases** in window: all q -permutations that contain start node;
- 5 **shift window** to the right by one node;
- 6 if *window extends the right border* then **wrap window**;
- 7 until *processed all window positions*

- Example: **stem**, **sorted sibling sequence**, **window** $w = 3$



Solution: Windowed pq -Gram Bases

Algorithm 1: **Form bases** from a sorted sibling sequence

- 1 if *sibling sequence* $<$ *window* then **extend with dummy nodes**;
- 2 **initialize window**: start with leftmost node;
- 3 repeat
- 4 **form bases** in window: all q -permutations that contain start node;
- 5 **shift window** to the right by one node;
- 6 if *window extends the right border* then **wrap window**;
- 7 until *processed all window positions*

- Example: **stem**, **sorted sibling sequence**, **window** $w = 3$

