Pattern Matching

Boyer Moore Algorithm

Definition 0.1

Last Occurrence Table: records the index of the last occurrence of the letter. We store it in a pair <letter, index> in a hashmap, and letters not in the alphabet of the pattern as marked as null, or returned as −1 in the functionality

Boyer Moore Last Table(pattern)

```
m = pattern.length
last = HashMap<character, index>
for all i from 0 to m-1
  last = put(pattern[i], i)
end for
return last
```

Theorem 0.1

Actual Search Algorithm

1. Create the LSOT to optimize shifts past mismatches
2. Move right to left in pattern
3. If there is a match, continue comparing text and pattern
4. If there is a mismatch, look to see if text character is in the alphabet
   - If the char is in the alphabet, align them
   - If the char is not in the alphabet, then shift past mismatched area altogether

Galil Rule

Definition 0.2

Galil Rule is a modification of Boyer-Moore after a complete match

- The Galil Rule improves on the individual shifts by performing an intelligent shift of the pattern after a complete match, helping approach linear time

The **period** of a string s1 is defined as the *shortest* prefix of s1 such that if we were to form a new string s2 by repeating this prefix, then s1 would be a prefix of s2.

When a *full match is found*, the Galil Rule exploits the period of the pattern to avoid unnecessary comparisons.