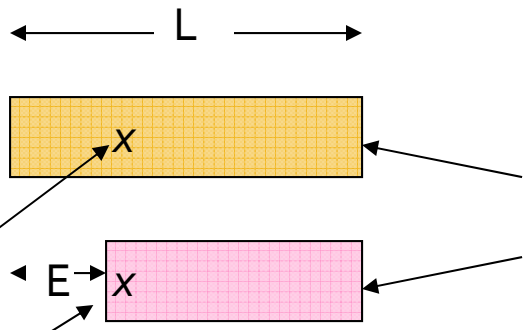
The Prefix of a String

Indexing by Symbols
Prefixes

Example: Prefix-Based Indexing

- If two strings are 90% similar, they must share some symbol in their prefixes whose length is just above 10% of the length of each string.
- Thus, we can base an index on symbols in just the first $\lfloor JL+1 \rfloor$ positions of a string of length L.
  - That's the *prefix* of the string.

Why the Limit on Prefixes?



Must be Equal

Extreme case: second string has none of the first E symbols of the first string, but they agree thereafter.

If two strings do not share any of the first E symbols, then $J \geq E/L$.

Thus, E = JL is possible, but any larger E is impossible. Index E+1 positions.

Indexing Prefixes

- Think of a bucket for each possible symbol.
- Each string of length L is placed in the bucket for each of its first $\lfloor JL+1 \rfloor$ positions.
- A B-tree with symbol as key leads to the strings.

Lookup

- Given a *probe* string *s* of length L, with J the limit on Jaccard distance:

```
for (each symbol a among the
 first ⌊JL+1⌋ positions of s)
   look for other strings in
    the bucket for a;
```

Example: Indexing Prefixes

- Let J = 0.2.
- String abcdef is indexed under *a* and *b*.
- String acdfg is indexed under *a* and *c*.
- String bcde is indexed only under *b*.
- If we search for strings similar to cdef, we need look only in the bucket for *c*.