

If we re-start past the point of mis-match, we will have the following:

```
Text:      A B A B A B A B D X X X X X
Pattern:   A B A B D
           ^
           | Incorrect re-starting point.
```

In this case, the correct re-starting position is at the point of mis-match:

```
Text:      A B A B A B A B D X X X X X
Pattern:   A B A B D
           ^
           | Correct re-starting point, leading to a match.
```

Observe: Moving completely past the point of mis-match depends on the character in the text. At best, considering the information provided by the text character at the point of mis-match will move the search a bit further forward. However doing so adds additional processing steps to the matching algorithm. The Knuth-Morris-Pratt algorithm ignores the text character at the point of mis-match. Instead, it relies entirely on pre-processing the pattern to extract information to assist choosing the re-starting point. In doing so, the search code is simplified.

In addition, the Knuth-Morris-Pratt takes care that the re-starting position does not accidentally skip over a re-starting point where a match may occur.

Consider another example:

```
Text:      A B B A B B A B C X X X X X
Pattern:   A B B A B C
           ^
           | mis-match occurs here.
```

In this case, a prefix of the pattern (i.e., `A B`) matches a suffix of the pattern just prior to the pattern character (`'C'`). We can use this common prefix/suffix to move the pattern forward, and to restart at a new position partially into the pattern.

Notice that re-starting the search at the point of mis-match (`'C'`) in the text (and at the beginning of the pattern) is incorrect. Doing so would fail to find an obvious match starting in position 3 in the text.¹

In the example above, we know at the point of mis-match, that we have already successfully matched `'A B'`, just prior to not matching on `'C'`. Our pattern begins with `'A B'` and we use this information to restart the search. Consider the following diagram:

```
           | Pattern moves forward to this point.
           v
Text:      A B B A B B A B C X X X X X
Pattern:   A B B A B C
           ^
           | matching re-starts here.
```

¹For both the text and the pattern, we count positions in the string starting with position zero.