Sergio E. Garcia Tapia
Algorithms by Sedgewick and Wayne (4th edition) [SW11]
January 25, 2025

5.3: Substring Search

Notes

To construct the `dfa[][]` array that corresponds to a pattern, the idea is to first fill in the match transitions. Note that `dfa[pat.charAt(j)][j]` is always `j+1` because if we are in state `j` (we have seen `j` characters `pat.charAt(0..j-1)` in the pattern) and see `pat.charAt(j)`, then we matched. Then, the challenge is to understand the mismatch transitions. As explained in [SW11], the key is to track what state the DFA would be in if we had a mismatch and had to back up.

Say we mismatch when reading `txt.charAt(i)` while in state `j`. This means that the first `j` characters matched, but not the `j+1`-th character, so the substring does not match at position `i-j` of the text. Since we have seen `pat.charAt(0..j-1)` of the pattern in the text, we can use those characters to decide which state to backup to. If we were to back up in the brute-force method, we would move back to `txt.charAt(i-j+1)`, which corresponds to `pat.charAt(1)`. Then, we would check `pat.charAt(1..j-1)`, the pattern characters that we know occur in the text, and use them to figure out how where to restart.

Say we are in state `5` and encountered a mismatch with a character `c` that does not match `pat.charAt(5)`. If we backed up, we would process characters `pat.charAt(1..4)` in the DFA from the beginning, and end up at a state `x`, called the *restart state* `x` corresponding to state `5`. Now on processing `c` at state `5`, we transition to where the DFA would go when receiving `c` while in state `x`.

Exercises

Exercise 2. Give the `dfa[][]` array for the Knuth–Morris–Pratt algorithm for the pattern `A A A A A A A A A`, and draw the DFA, in the style of the figures in the text.

Solution. See Figure 1. Notice that receiving any character that is not `A` would reset us to state `0`.

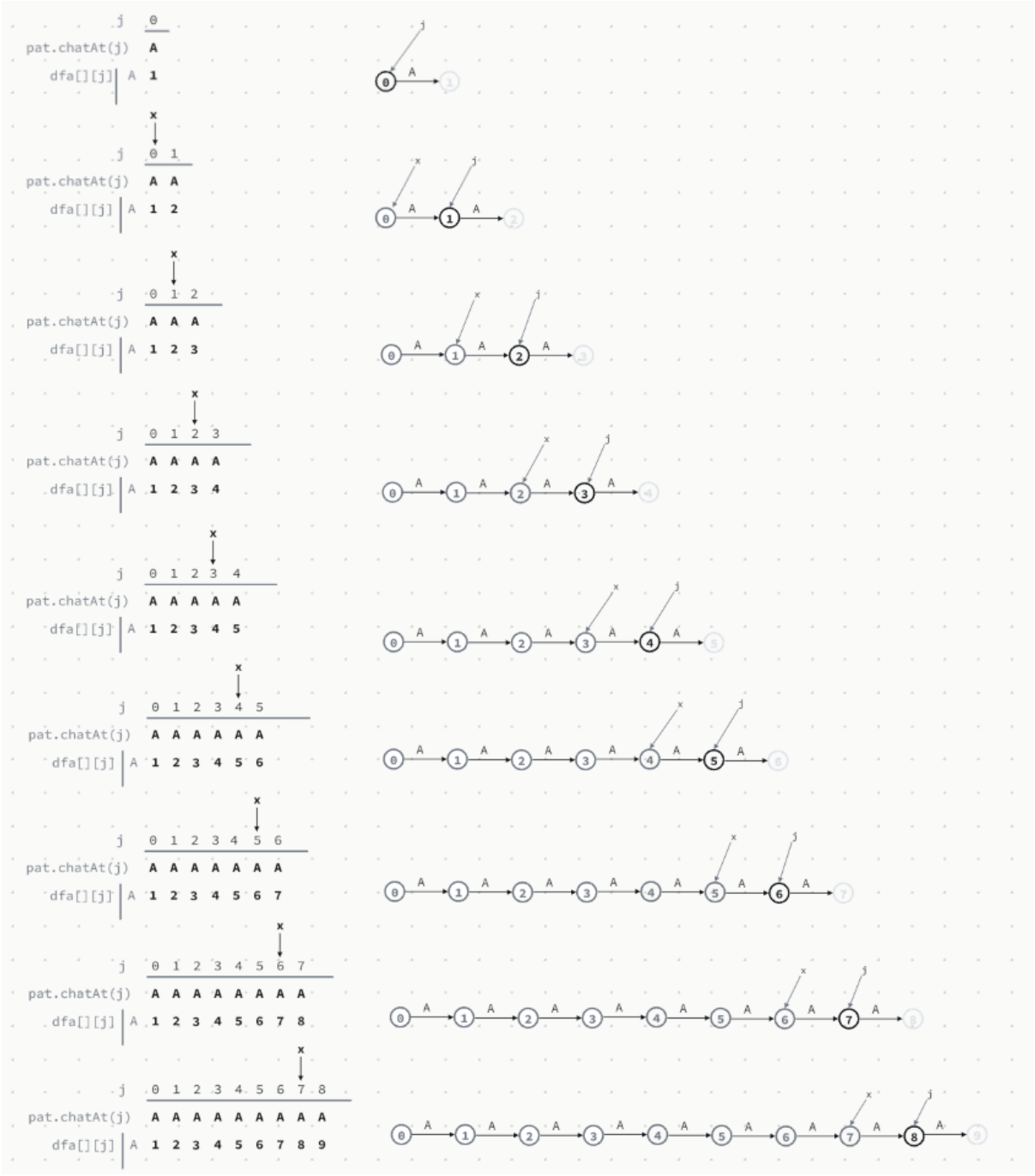

Figure 1: Constructing the DFA For KMP substring search in Exercise 2.

References

[SW11] Robert Sedgewick and Kevin Wayne. *Algorithms*. 4th ed. Addison-Wesley, 2011. ISBN: 9780321573513.