The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Solution Manual
Chapter 3, Problem 3.23

Intuitively, verifying membership in that language appears to require that we check all $n + 1$ decompositions of the input string z into substrings x and y with $z = xy$ and evaluate each substring according to the procedure described in Exercise 3.12 before testing two results for equality. The equality testing is not really a problem, since each expression must assume a value from the set $\{a, b, c\}$—we can just maintain in the finite-state control the values of the two expressions by turning the states into triples, with the first part of the triple corresponding to a state of the machine described in Exercise 3.12 and the next two parts corresponding to the two values. We can use nondeterminism to guess the correct breakdown of z into x and y, proceeding on x just as in Exercise 3.12, storing the computed value into the finite-state control, then proceeding on y again as in Exercise 3.12 (but now with states that store the value computer for x), and accepting if the value computed for y matches that computed for x.