11.串

(a) ADT

邓俊辉

deng@tsinghua.edu.cn

定义

❖ 由来自字母表Σ的字符所组成的有限序列：

T s i n g h u a

$$S = a_0\ a_1\ a_2\ \ldots\ a_{n-1} \in \Sigma^*$$

❖ 既然如此，为何不直接用序列来实现串？

❖ 通常，字符的种类不多，而串长 = n >> |Σ|

❖ 比如：

英文文章	(['A'-'Z'] ∪ ['a'-'z'] ∪ { ' ', '.' , ',' , ... })*
C/C++程序	({ 95个可打印字符 } ∪ { LF, CR })*
天然蛋白质	{ 21种氨基酸 }*
DNA	{ A, C, G, T }*
RNA	{ A, C, G, U }*
二进制	{ 0 , 1 }*

术语

- ❖ 相等：S[0, n) = T[0, m)

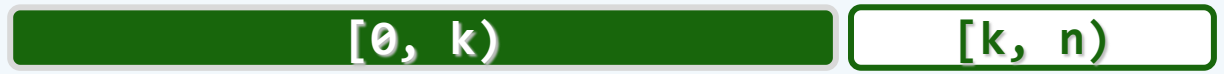
  长度相等（n = m），且对应的字符均相同（S[i] = T[i]）

- ❖ 子串：S.substr(i, k) = S[i, i + k), 0 ≤ i < n, 0 ≤ k

  亦即，从S[i]起的连续k个字符

- ❖ 前缀：S.prefix(k) = S.substr(0, k) = S[0, k), 0 ≤ k ≤ n

  亦即，S中最靠前的k个字符

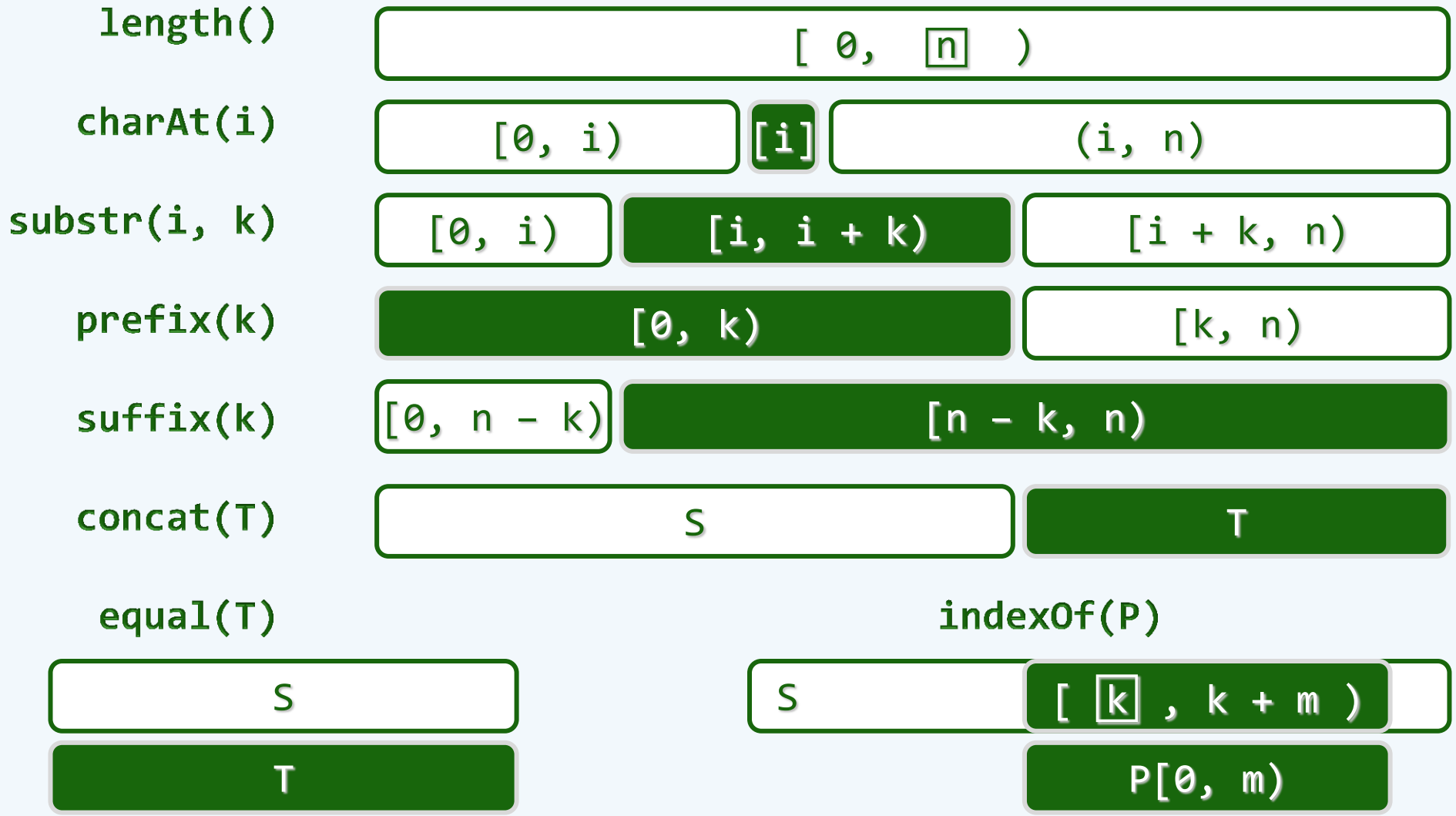


- ❖ 后缀：S.suffix(k) = S.substr(n - k, k) = S[n - k, n), 0 ≤ k ≤ n

  亦即，S中最靠后的k个字符

- ❖ 联系：S.substr(i, k) = S.prefix(i + k).suffix(k)
- ❖ 空串：S[0, n = 0)，也是任何串的子串、前缀、后缀

ADT


length()
[0, n)
charAt(i)
[0, i)
[i]
(i, n)
substr(i, k)
[0, i)
[i, i + k)
[i + k, n)
prefix(k)
[0, k)
[k, n)
suffix(k)
[0, n – k)
[n – k, n)
concat(T)
S
T
equal(T)
S
T
indexOf(P)
S
[k , k + m)
P[0, m)

实例

- "data structures".length() = 15

  "data structures".charAt(5) = 's'

  "data structures".prefix(4) = "data"

  "data structures".suffix(10) = "structures"

  "data structures".concat(" & algorithms") = "data structures & algorithms"

  "algorithms".equal("data structures") = false

  "data structures and algorithms".indexOf("string") = -1

  "data structures and algorithms".indexOf("algorithm") = 20

- <string.h>中的对应功能：strlen()、strcpy()、strcat()、strcmp()、strstr()

- 以下，直接利用字符数组实现字符串，转而重点讨论串匹配算法