



۱ کاربرد شباهت بین دو رشته

۱.۱ تصحیح املاء

پیدا کردن کلمه مناسب نوشتاری با درست ترین فاصله نگارشی

۲.۱ بیوانفورماتیک: مقایسه رشته های زیستی

پیدا کردن شباهت دو موجود با محاسبه فاصله ی دو ژنوم

۳.۱ ارزیابی ترجمه ماشینی یا تشخیص گفتار

پیدا کردن شباهت و تفاوت دو جمله تولید شده با ماشین و تولید شده از مرجع

۲ فاصله ویرایشی

فاصله کمینه ویرایشی بین دو رشته که با کمترین تعداد عملگرهای زیر برای تبدیل به رشته دیگر انجام میشود.

- اضافه شدن: حرفی در رشته نباشد که در رشته دیگر باشد.
- حذف شدن : حرفی در رشته باشد که در رشته دیگر نباشد.
- تغییر حرف: ۲ حرف متفاوت باشند.

۱.۲ پیدا کردن فاصله کمینه ویرایشی

- شروع: رشته اولیه  $X \leq n$  به طول n
- عملگرها: تولید، حذف، تبدیل
- هدف: رشته ثانویه  $Y \leq m$  به طول m
- تابع هدف: کمینه کردن هزینه‌ها  $D(i,j) \leq$

۲.۲ الگوریتم یافتن فاصله مینم ویرایشی

۲ رشته‌ی مورد نظر را در پایین و چپ جدول وارد میکنیم و از ۱ تا طول رشته مقادیر اولیه میدهیم. از پایین چپ جدول شروع کرده خانه های بالا و راست را ۱+ مقداردهی میکنیم. خانه بالا راست را اگر هر دو حرف سطر و ستون آن یکی بودند ۰+ و در غیر اینصورت ۲+ میدهیم. راست بالاترین خانه کمترین فاصله ویرایشی و مسیر طی شده کمترین مسیر است.

Initialization:

$$D(i,0) = i$$

$$D(0,j) = j$$

Recurrence Relation:

For each i = 1 to M

For each j = 1 to N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } X(i) \neq Y(j) \\ 0; & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

Termination:

$D(N,M)$  is distance

۳.۲ عملکرد الگوریتم

زمان:  $O(nm)$

حافظه:  $O(nm)$

ردیابی:  $O(n+m)$

۴.۲ مثال

فاصله ویرایشی دو کلمه‌ی intention و execution را محاسبه میکنیم.

مشابه الگوریتم گفته شده عمل میکنیم و مقداردهی میکنیم.

N	9										
O	8										
I	7										
T	6										
N	5										
E	4										
T	3										
N	2										
I	1										
#	0	1	2	3	4	5	6	7	8	9	
	#	E	X	E	C	U	T	I	O	N	

با توجه به حروف کلمه از پایین چپ پایین به بالا راست حرکت میکنیم. بالا راست‌ترین خانه جواب است.

N	9	8	9	10	11	12	11	10	9	8
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
T	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

## ۵.۲ میزان اشتباه جایگزین شدن حروف

- محل قرارگیری حروف در کیبورد
- هزینه حروف نزدیک را کمتر قرار میدهیم.
- شکل و تلفظ حروف
- هزینه حروف هم آوا و یا نوشتار مشابه را کمتر قرار میدهیم.

## ۶.۲ محاسبه فاصله کمینه وزندار

با توجه به ارزشی که به هر عملگر داده میشود مقدار بدست می‌آید.

Initialization:

$$D(0,0) = 0$$

$$D(i,0) = D(i-1,0) + del[x(i)]; 1 < i <= N$$

$$D(0,j) = D(0,j-1) + ins[y(j)]; 1 < j <= M$$

Recurrence Relation:

$$D(i-1,j) + del[x(i)]$$

$$D(i,j)= \min D(i,j-1) + ins[y(j)]$$

$$D(i-1,j-1) + sub[x(i),y(j)]$$

Termination:

$$D(N,M) \text{ is distance}$$

## ۳ زبانهاي منظم

L یک زبان منظم است اگر و فقط اگر توسط یک DFA یا NFA پذیرفته شود.

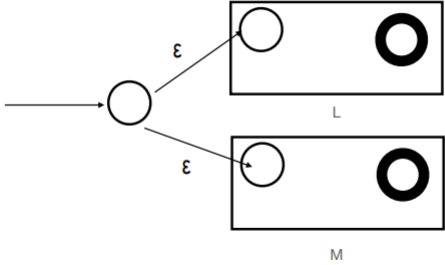
M یک automaton finite deterministic ۵ تایی شامل  $(Q, \sigma, \delta, q^{\circ}, F)$  است که:

- مجموعه‌ای محدود از حالات Q
- مجموعه‌ای محدود از نمادهای ورودی به نام الفبای  $\sigma$
- یک تابع انتقال  $Q * \sigma => Q$
- حالت اولیه یا شروع
- مجموعه‌ای از حالات پذیرفته شده

## ۱.۳ عملگرهای یک زبان منظم

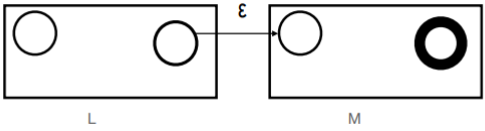
- Union

$$L \cup M$$



- Dot

$$M \bullet L$$



- Star

بسته شدن یک زبان،  $L^*$ ، به صورت  $L^{\circ} \cup L_1 \cup L_2 \cup L_3 \cup ...$  تعریف می شود که در آن  $L^{\circ} = \epsilon$  و  $L_1 = L$



## ۲.۳ گرامر زبان منظم

خطی از راست است که نحوه‌ی تولید و شکل رشته را مشخص میکند.

خطی: سمت چپ فلش تنها یک نانترمینال داریم.

خطی از راست: ترمینال‌ها سمت چپ نانترمینال‌ها قرار میگیرند.

