

## تمرین دوم اتوماتا

نیما بهرنگ ۹۶۱۰۰۱۱۴

۲۵ آبان ۱۳۹۷

استاد خزایی

پرسش ۱

## پرسش ۲

چون در تمام قسمت ها غیر از قسمت ث از لم تزریق استفاده می کنیم، فرض این لم را در این جا می نویسیم و در هر قسمت مستقیم به طرح روش ۱ ساخت طبق لم تزریق می پردازیم  
طبق لم تزریق اگر به هر ازای  $n$  استیت اتوماتای محدود، وجود داشته باشد کلمه ای که تعداد حرف های آن از  $n$  بیشتر باشد و

$$\forall x, y, z \in 0, 1^* : \left\{ \begin{array}{l} w = xyz \\ |xy| \leq n \\ |y| \geq 1 \end{array} \right\} \Rightarrow \exists i \geq 0 : xy^i z \notin L$$

$$w = 0^n 1^n 2^n \Rightarrow y = 0^k, |y| \geq 1 \Rightarrow xy^2 z = 0^{n+k} 1^n 2^n \notin L \quad ( \text{ا} )$$

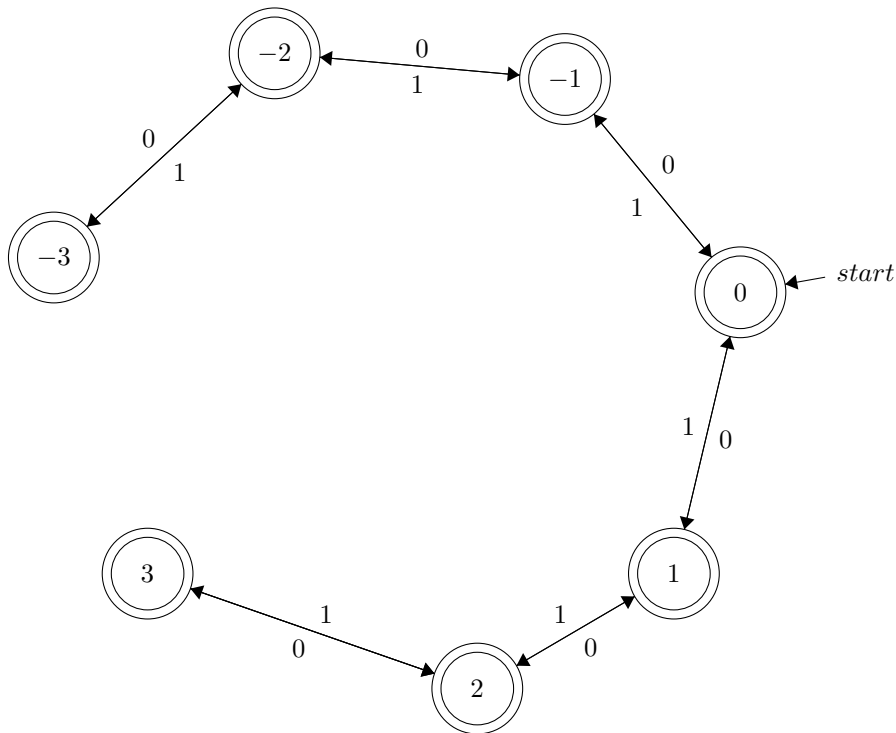
$$w = 0^n 1^{2n} 2^n \Rightarrow y = 0^k, |y| \geq 1 \Rightarrow xy^2 z = 0^{n+k} 1^{2n} 2^n \notin L \quad ( \text{ب} )$$

$$w = 0^n 1^{2n} 2^n \Rightarrow y = 0^k, |y| \geq 1 \Rightarrow xy^2 z = 0^{n+k} 1^{2n} 2^n \notin L \quad ( \text{ج} )$$

$$w = 0^{2n-1} 1^n \Rightarrow y = 0^k, k \geq 1, |y| \geq 1 \Rightarrow xy^2 z = 0^{2n-1+k} 1^n, 2n-1+k \geq 2 * n \Rightarrow w \notin L \quad ( \text{د} )$$

$$w = 0^n 110^n \Rightarrow y = 0^k, |y| \geq 1 \Rightarrow xy^2 z = 0^{n+k} 110^n \Rightarrow w \notin L \quad ( \text{ه} )$$

و ( ماشینی ارائه می دهیم که هر استیت آن نشان دهنده تفاضل یک ها از صفر ها باشد و طبق فرض مسئله نباشد بیشتر از ۳ یا کمتر از -۳ شود و در ضمن با انتخاب یک یا صفر از هر استیت به استیتی متناظر تفاضل جدید می رویم تمام استیت ها باید فاینال باشد زیرا هیچ زمانی نباید فرض مسئله نقض شود و همچنین اگر فرض مسئله رعایت شده باشد به آن معنی است که به یک حالت نهایی رفته ایم



$$w = 0^n 1 2^n \Rightarrow y = 0^k, k \geq 1, |y| \geq 1 \Rightarrow xy^2z = 0^{n+k} 1 2^n \Rightarrow (n+k) * 1 \neq ( \ ;$$

$$n \Rightarrow w \notin L$$

### پرسش ۳

$$\begin{aligned} \epsilon &\in (0+01)^* \quad \text{آ} \\ \epsilon &\notin 00^*1^* \end{aligned}$$

$$\begin{aligned} \epsilon &\in (0+1)^*(11+0+\epsilon)^* \quad \text{ب} \\ \epsilon &\notin (0+1)^*11(0+1)^* \end{aligned}$$

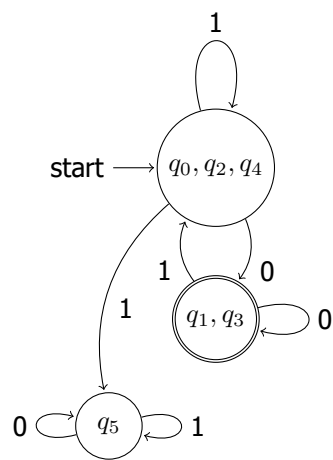
$$\begin{aligned} 1 &\in (1+10+11+1)^* \quad \text{ج} \\ 1 &\notin (111+110+101+011+010+0)^* \end{aligned}$$

د برابرند زیرا در زبان هر جفتشان هر کدام از 00, 01, 10, 11 به دلخواه می تواند باشد همچنین دی اف ای ضربیشان نیز تهی است زیرا هر حالت دی اف ای شان می توان یکی از 00, 01, 10, 11 را انتخاب کرد و برای هر دو امکان پذیر است و بعد از هر انتخاب نیز به حالتی نهایی می روند

# پرسش ۴

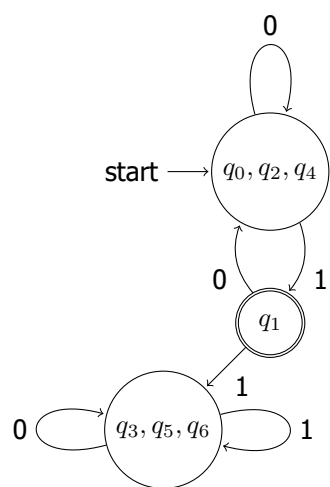
( آ )

	$q_0$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$
$q_0$		X		X		X
$q_1$	X		X		X	X
$q_2$		X		X		X
$q_3$	X		X		X	X
$q_4$		X		X		X
$q_5$	X	X	X	X	X	



( ب )

	$q_0$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$
$q_0$		X		X		X	X
$q_1$	X		X	X	X	X	X
$q_2$		X		X		X	X
$q_3$	X	X	X		X		
$q_4$		X		X		X	X
$q_5$	X	X	X		X		
$q_6$	X	X	X		X		



## پرسش ۵

آ ( اگر رشته های  $0^k 1, k \geq 1$  را در نظر بگیریم، دارای خاصیت گفته شده هستند، زیرا هر دو عضو آن را که بگیری و متفاوت باشند مثلا  $0^n 1, 0^m 1, n \neq m$  تنها رشته ای که به هر کدام از آنها می تواند اضافه شود و یک رشته عضو زبان بسازد رشته  $1^{k-1}$  است که همواره به ازای هر دو عضو متفاوت و یکتاست پس یک مجموعه مورد نظر است که نامتناهی هم هست )

## پرسش ۶

```
<pair<char, int>> reverseAdj[n][k] \\ reverse edges in automata
\\ like if there was Q (a)-> P now : P (a)-> Q
vector<int> finals;
boolean isfinal[n];
int start;
boolean difftable[n][n]

void notfinal(int p, int q){
    difftable[p][q] = difftable[q][p] = true;
    for(int i = 0; i < k; i++){
        p2 = reverseAdj[p][i];
        q2 = reverseAdj[q][i];
        if(!difftable[p2][q2])
            notfinal[p2][q2];
    }
}

int main(){
    for(int i = 0; i < finals.size(); i++)
        isfinal[finals[i]] = true;
    for(int i = 0; i < finals.size(); i++)
        for(int j = 0; j < n; j++)
            if(!final[j] && !difftable[final[i]][j])
                notfinal(final[i], j)
}
```

الگوریتم گفته شده همانطور که در کلاس گفته شد بر اساس اینکه هر فایلال با هر غیرفاینال، دو استتیت متمایزند، در الگوریتم ما نیز همینگونه شروع می کند و به صورت بازگشتی روی قبلی ها اجرا می شود  
 آوردن برنامه برابر  $N + O(\text{notfinal}())$  است که تعداد دفعاتی که تابع ما صدا می شود برابر مربع تعداد استتیت هاست زیرا هر زوج مرتب تنها حداکثر یک بار صدا زده می شود و آن هم وقتی است که زوج مرتب دیگری که برای اولین بار صدا شده اند، با یکی از حروف زبان به این استتیت آمده باشند، و سپس این زوج علامتشان زده می شود و هیچ بار دیگری تابع به ازای آنها صدا نمی شود پس تابع ما حداکثر به تعداد مربع استتیت ها می تواند صدا شود  
 پس آوردن برنامه همان مربع استتیت است.



## پرسش ۷

چون هر زبان منظم يك عبارت منظم دارد و طبق گفته های جزوه ۷ استاد می توان هر يك از عملیات الحاق و اجتماع و بستار را با يك  $\epsilon - nfa$  كه تنها يك استیت فاینال دارد ساخت پس هر زبان منظم نیز يك  $\epsilon - nfa$  با تنها يك استیت پایان دارد

پرسش ۸

پرسش ۹