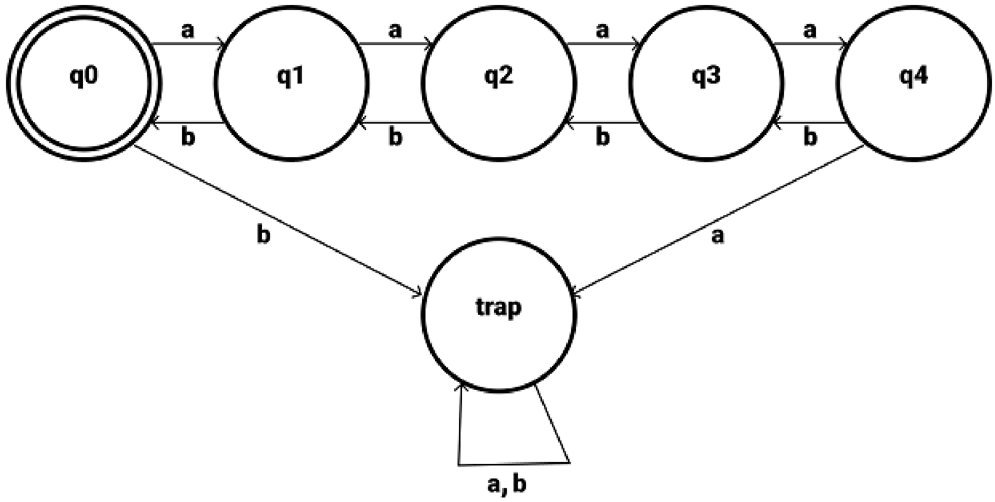
**تمرین ۲** **ایمان محمدی ــ ۹۹۱۰۲۲۰۷**

**نیم‌سال دوم ۱۴۰۱ــ۱۴۰۲** **زمان آپلود: ۱۱ اردیبهشت**

**سوال ۱.**

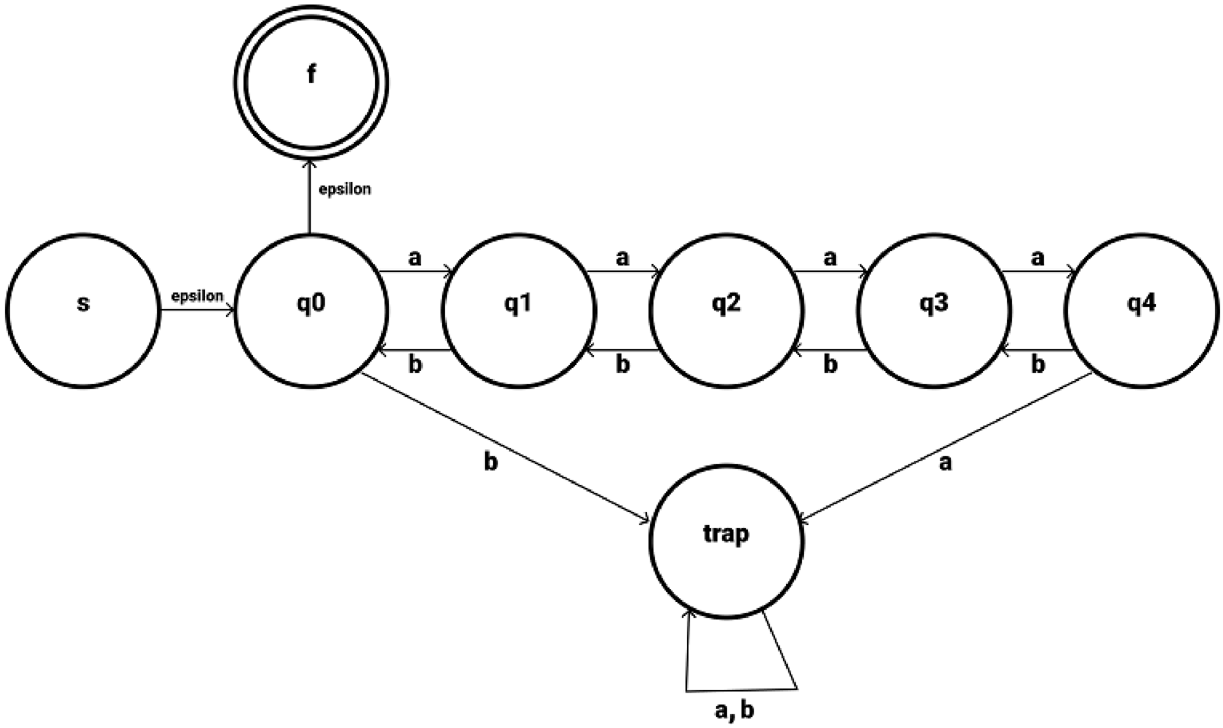
### ۱.۱

در ابتدا، DFA این زبان را رسم می‌کنیم. برای این‌که از هرگونه اشتباهی جلوگیری کنیم، پرانتز باز "(" را با a نشان می‌دهیم و پرانتز بسته ")" را با b نشان می‌دهیم. چون هیچ پرانتز باز و بسته ای وجود ندارد یا در واقع تعداد هر دو ۰ است، به عنوان state accept آن‌ها را می‌پذیریم. در استیت اولیه، با دیدن پرانتز بسته به حالت trap رفته و با دیدن پرانتز باز، به استیت بعدی می‌رویم. همچنین در استیت‌های بعدی، با دیدن پرانتز بسته به استیت قبلی رفته و با دیدن پرانتز باز به حالت بعدی می‌رویم. همچنین اگر تعداد پرانتزهای باز بسته نشده به بیش از ۴ عدد برسد، به حالت trap می‌رویم. همچنین همان استیت اولیه‌ی ما، استیت accept می‌باشد.

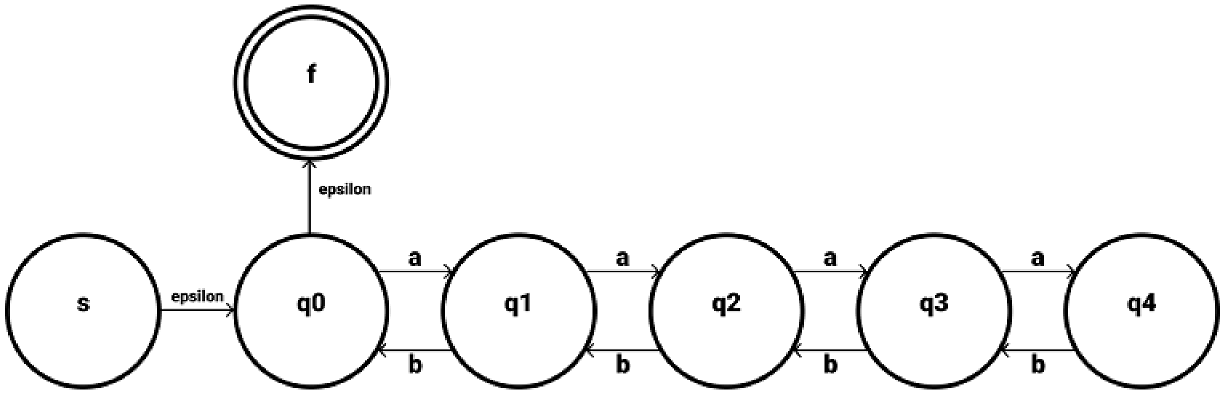



حالا برای تبدیل آن به عبارات منظم، باید به شکلی از GNFA در آوریم آن را. در نتیجه باید یال ورودی به استیت ۰ و یال خروجی از استیت پایانی نداشته باشیم. همچنین این ۲ استیت باید متفاوت باشند و مثل DFA کشیده شده نباید یکسان باشند. برای این‌کار، استیت s را به عنوان استیت شروع اضافه می‌کنیم به DFA کشیده شده و از استیت پایانی آن را جدا می‌کنیم. از استیت s به استیت شروع قبلی یک یال اپسیلون اضافه می‌کنیم. همچنین استیت f اضافه می‌کنیم به DFA و از استیت پایان قبلی یک یال اپسیلون به آن اضافه می‌کنیم. حالا بین هر ۲ حالتی که یال نداریم باید یال‌های تهی اضافه کنیم. (این یال‌ها در عکس پایین نشان داده نشده‌اند ولی)

در نتیجه الان استیت شروع و پایان جدیدی داریم که یکسان نیستند و یال ورودی به استیت شروع و یال خروجی از استیت پایانی نداریم.



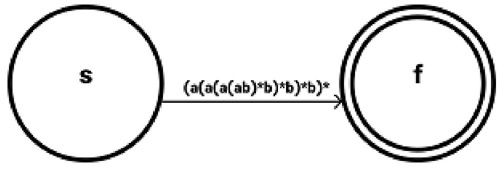

حالا مرحله به مرحله به حذف یال‌ها می‌پردازیم و GNFA را آپدیت می‌کنیم. با توجه به این‌که حالت trap فقط به خودش یالِ غیر از تهی دارد، با حذف آن تغییری در باقی استیت‌ها و سایر یال‌ها رخ نمی‌دهد و چون concat یک string با $\emptyset$، خود $\emptyset$ می‌شود، تغییری به طور کلی ایجاد نمی‌شود با حذف آن.




حالا به ترتیب، به حذف استیت‌های ۴، ۳، ۲، ۱ و ۰ می‌پردازیم تا فقط استیت‌های s و f باقی بمانند. برای هر کدوم از آن‌ها بررسی می‌کنیم که چه یال‌های غیرتهی دارند به دیگر استیت‌ها و مطابق الگوریتم گفته شده، به خودشان وصل می‌کنیم استیت‌های قبلی استیت‌های حذف شده را و به ترتیب که این‌کار را انجام دهیم، DFA مدنظر ما بدست می‌آید. در نهایت با توجه به این روش و روند حل، در نهایت عبارت منظم خواسته شده به شکل

$$(a(a(a(ab)*b)*b)*b)*$$

می‌باشد.



دلیل درستی این DFA این است که به ازای هر ورودی پرانتز می‌بایست یک ورودی پرانتز بسته یا y دریافت کنیم که به استیت‌های قبلی و در آخر به استیت شروع که حالت accept ما می‌باشد، برویم. همچنین اگر عمق بیش از ۴ شود، به استیت trap می‌رویم و در آن می‌مانیم همیشه.

## ۱.۲

در بخش دوم از سوال ۱ ، تابع stammer به صورت بازگشتی تعریف شده است و هر حرف رشته‌ی ورودی خودش را، دو بار تکرار می‌کند. برای مثال در صورتی که ورودی آن $w_1w_2...w_n$ باشد، خروجی آن $w_1w_1w_2w_2...w_nw_n$ خواهد بود. حالا دستگاهی را در نظر می‌گیریم که زبان L را می‌شناسد؛ می‌توان دستگاه جدیدی ساخت که زبان $stammer^{-1}(L)$ را بشناسد. برای این دستگاه خواهیم داشت

زبان L :

$$M = (Q, \Sigma, \delta, q, F)$$

زبان $stammer^{-1}(L)$

$$N = (Q, \Sigma, \delta_2, q, F)$$

برای قواعد زبان $stammer^{-1}(L)$ داریم که

$$\delta_2(q, a) = \delta(\delta(q, a), a)$$

پس این ماشین می‌تواند زبان مدنظر را شناسایی کند و اگر در ماشین ابتدایی ورودی $w_1w_1w_2w_2...w_nw_n$ داشته باشیم، از استیت‌های $q_1q_2...q_n$ عبور می‌کند و در ماشین جدید، رشته‌ی $w_1w_2...w_n$ از استیت‌های فرد یعنی $q_1q_3q_5...$ عبور می‌کند. پس چون توانستیم برای این زبان، DFA طراحی کنیم، این زبان منظم است.

## ۱.۳

در این بخش از یک سمت تساوی شروع می‌کنیم و به سمت دیگر می‌رسیم.

$$(xy^*z)^*(xyz(xy^*z)^*)^* = (xy^*z \cup xyz)^*, \quad \text{با توجه به همانی پایه } (xy^*z)^* = (x \cup y)^*$$

$$= (x(y^* \cup y)z)^*, \quad \text{با توجه به همانی پایه } L(M \cup N) = LM \cup LN,\ (M \cup N)L = ML \cup MN$$

$$= (xy^*z)^*, \quad \text{با توجه به } y^* = (y^* \cup y)$$

با توجه به این‌که اجتماع رشته $y$ و $y^*$ می‌شود $y^*$ ، چون که از صفر یا بیشتر $y$ تشکیل شده است.

برای قسمت بعدی هم ابتدا نشان می‌دهیم.

$$(x \cup y)^* = x^*(yx^*)^*$$

می‌دانیم رشته $x^*(yx^*)^*$ زیرمجموعه‌ی رشته $(x \cup y)^*$ است زیرا که تنها می‌تواند از حروف $x$ و $y$ تشکیل شده باشد. همچنین می‌دانیم $(x \cup y)^*$ زیرمجموعه $x^*(yx^*)^*$ است چرا که برای هر رشته متشکل از تعدادی $x$ و $y$ دلخواه از شمردن $x$ شروع می‌کنیم و هرجا $y$ بود، $yx^*$ را قرار می‌دهیم و تا انتها پیش می‌بریم. به این ترتیب هر رشته متشکل از این دو حرف در عبارت $x^*(yx^*)^*$ وجود خواهند داشت پس داریم.

$$(x \cup y)^* = x^*(yx^*)^*$$

برای حالت $(x \cup y)^* = x^*(x^*y)^*$ نیز به همین صورت خواهد بود با این تفاوت که این بار می‌توان از آخر رشته شروع کرد و هر $y$ دیدیم به همراه رشته $x$ به دنبال آن عبارت $x^*y$ را قرار دهیم. در نتیجه بدست می‌آوریم که:

$$(x \cup y)^* = x^*(x^*y)^* = x^*(yx^*)^*$$

---

**سوال ۲.** سوال ۲

## ۲.۱

برای ماشین متناهی غیرقطعی داده شده، ماشین متناهی قطعی‌ای طراحی کردیم که پذیره‌ی همان زبان است صرفا. (رشته‌های عضو زبان را می‌پذیرد و رشته‌های غیرعضو زبان را نمی‌پذیرد.)

در ابتدا استیت‌ها و تابع انتقال DFA مذکور را می‌نویسیم.

فرض می‌کنیم $L_۱$ زبانی می‌باشد که حتما با a شروع می‌شود، تعداد a و b هایش برابر است و اختلاف a و b تا هر جایی حداکثر ۱ است، از اول تا هیچ‌جای دیگری به جز آخر، تعداد a و b برابر نمی‌شود.

همچنین فرض می‌کنیم $L_۲$ زبانی می‌باشد که حتما با b شروع می‌شود، تعداد a و b هایش برابر است و اختلاف a و b تا هر جایی حداکثر ۱ است، از اول تا هیچ‌جای دیگری به جز آخر، تعداد a و b برابر نمی‌شود.

پس داریم:

$$L = (L_۱ \cup L_۲)^*$$

پس بدست می‌آید که:

$$L_۱ = ab, L_۲ = ba$$

حالا اگر زبان $L_۳$ را به این شکل تعریف کنیم که زبانی می‌باشد که حتما با a شروع می‌شود، تعداد a و b هایش برابر است و اختلاف a و b تا هر جایی حداکثر ۲ است، از اول تا هیچ‌جای دیگری به جز آخر، تعداد a و b برابر نمی‌شود.

همچنین زبان $L_۴$ را به این شکل تعریف کنیم که زبانی می‌باشد که حتما با b شروع می‌شود، تعداد a و b هایش برابر است و اختلاف a و b تا هر جایی حداکثر ۲ است، از اول تا هیچ‌جای دیگری به جز آخر، تعداد a و b برابر نمی‌شود.

به طور بازگشتی بدست می‌آوریم که:

$$L_۳ = a(ab)^*b$$

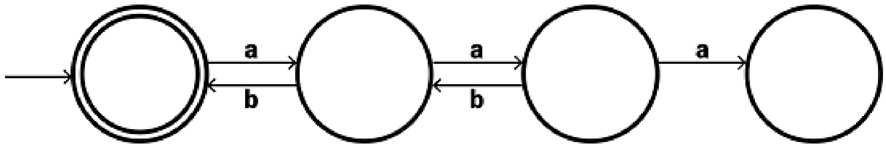
$$L_۴ = b(ba)^*a$$

پس در نهایت برای تعریف L داریم:

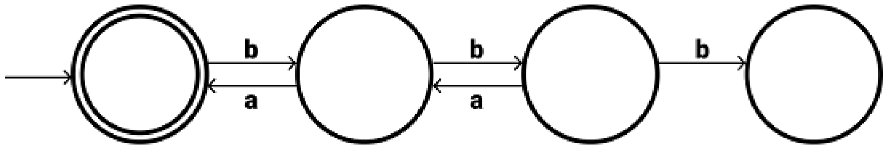
$$L = (a(ab)^*b \cup b(ba)^*a)^*$$

## ۲.۲

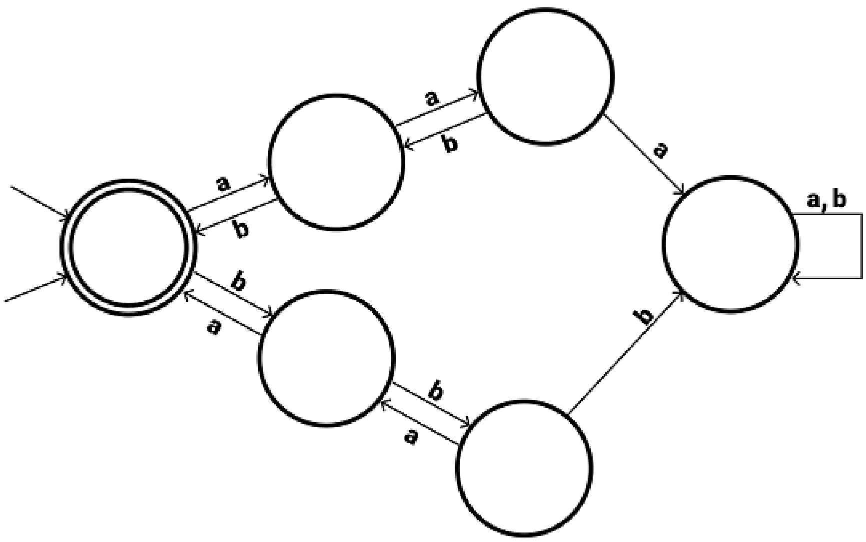
ماشین قطعی مربوط به هر کدام از زبان‌ها را حالا رسم می‌کنیم. برای زبان $L_۳$ خواهیم داشت:




همچنین برای زبان $L_۴$ خواهیم داشت:

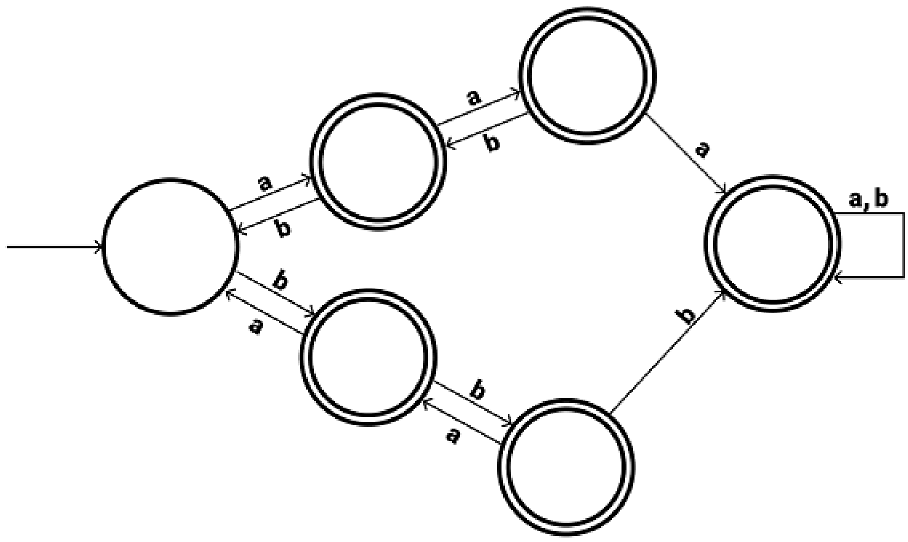



حالا با ترکیب این دو DFA ، ماشین قطعی برای کل عبارت منظم را بدست می‌آوریم:

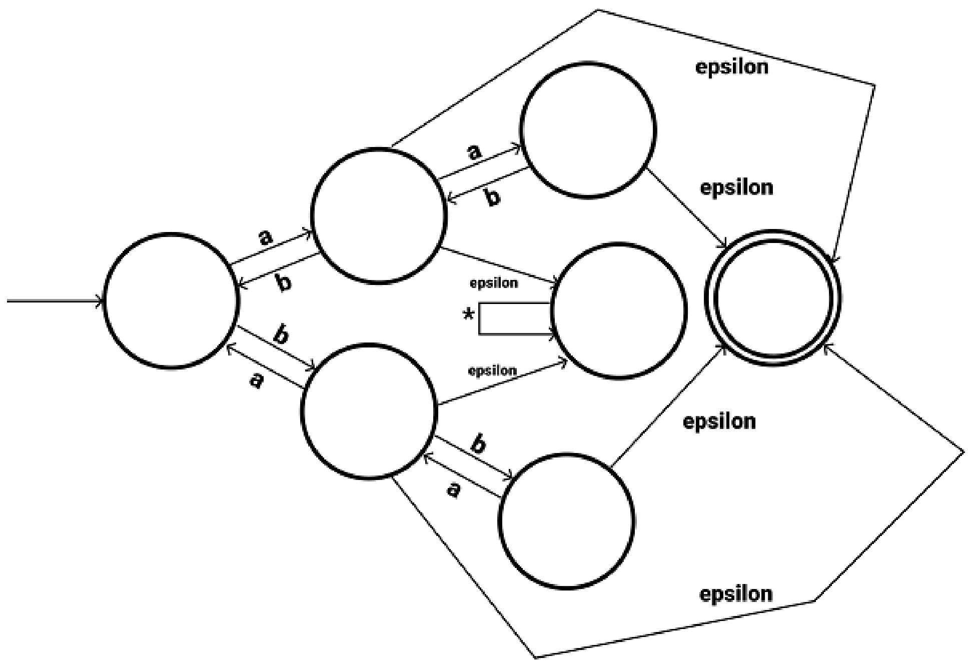



## ۲.۳

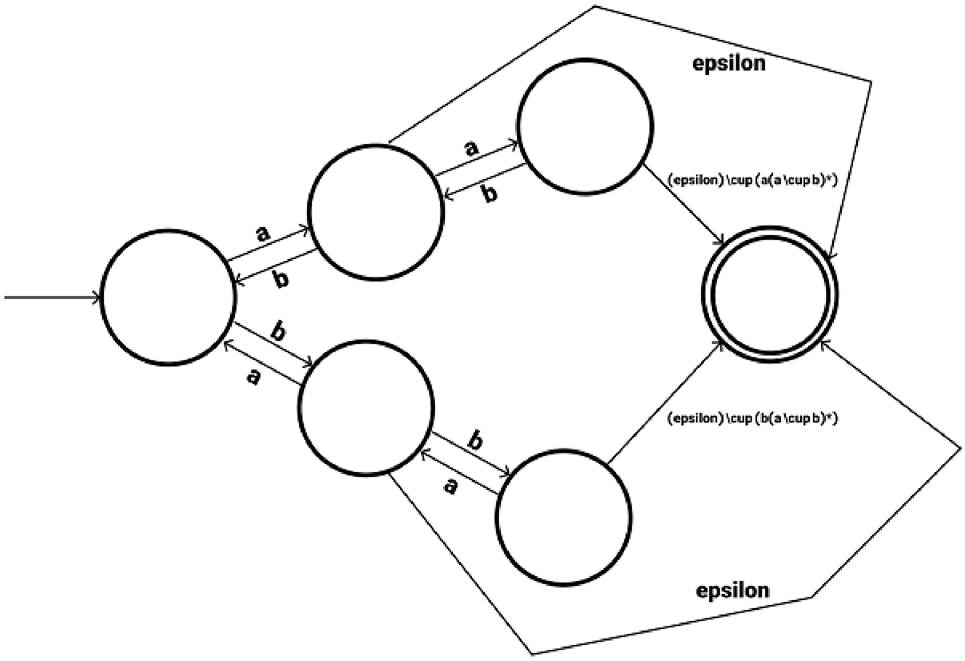
برای بدست آوردن ماشین مکمل ماشین موردنظر، حالات accept و reject را جابه‌جا می‌کنیم. در نتیجه مایشن مکمل می‌شود:



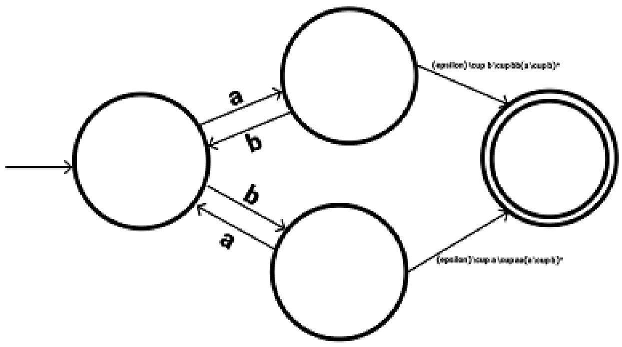

حالا برای عبارت منظم، اول این DFA را به GNFA تبدیل می‌کنیم. تفاوت این است که این‌جا باید یک استیت accept باید داشته باشیم، در نتیجه استیت‌های accept این DFA را با یک یال اپسیلون به یک استیت جدید وصل می‌کنیم. اسم آن استیت رو accept می‌گذاریم و این NFA بدست می‌آید:



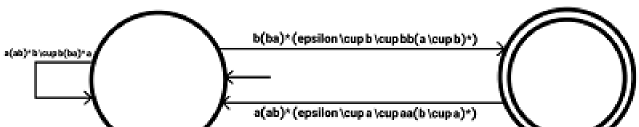

حالا تبدیل GNFA به عبارت منظم را انجام می‌دهیم:



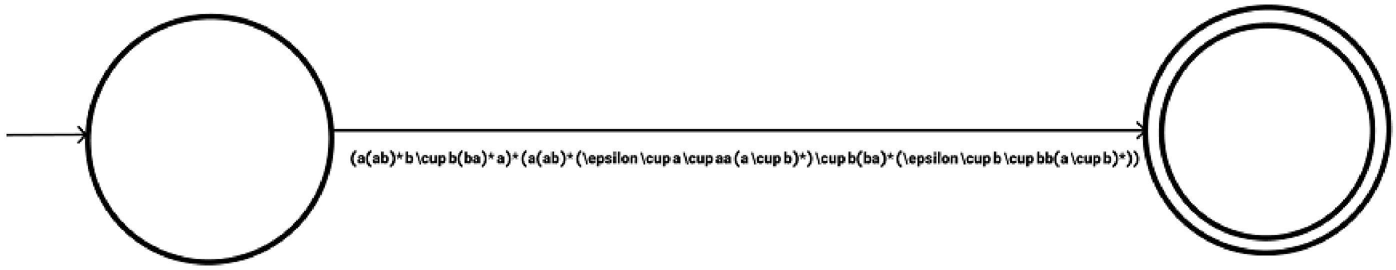

حالا ۲ راس پایینی را حذف می‌کنیم.




همچنین ۲ راس دیگر را نیز حذف می‌کنیم.




و در نهایت داریم:




پس در نهایت به DFA ای می‌رسیم با یالِ

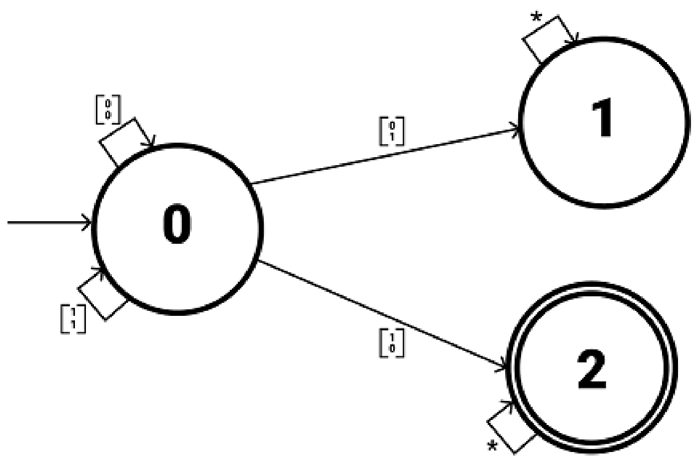
$$(a(ab)^*b \cup b(ba)^*a)^*(a(ab)^*(\epsilon \cup a \cup aa(a \cup b)^*) \cup b(ba)^*(\epsilon \cup b \cup bb(a \cup b)^*))$$

---

**سوال ۳.**

### ۳.۱

زبان $L_{۱}$ یک زبان منظم می‌باشد چون می‌توانیم برای آن یک dfa به صورت زیر طراحی کنیم.




به ازای دریافت حالت $\binom{۱}{۰}$ می‌دانیم عدد بالایی بزرگتر خواهد بود و برای $\binom{۰}{۱}$ عدد پایینی. در مابقی حالات در وضعیت خود می‌مانیم.

برای زبان $L_{۲}$ می‌دانیم هر رشته‌ای که در بالا تکرار شود معکوس همان رشته می‌بایست در پایین تکرار شود. بنابراین یک حالت آن می‌تواند به صورت زیر باشد.

$$\begin{bmatrix} ۱^p & ۰^p \\ ۰^p & ۱^p \end{bmatrix} = xyz$$

که در آن $|xy| \leq p$ و $|y| > ۰$ خواهند بود. پس می‌توان گفت $y$ به صورت $\binom{۱^k}{۰^k}, ۰ < k < p$ خواهد بود.

حالا در رشته‌ی می‌بینیم $xy^۲z$ که به $\begin{bmatrix} ۱^{p+k} & ۰^{p+k} \\ ۰^p & ۱^p \end{bmatrix}$ برمی‌خوریم، بنابراین این زبان را نمی‌توان با DFA خاصی توصیف کرد و یک زبان نامنظم است.

## ۳.۲

با استفاده از لم پمپاژ اثبات می‌کنیم که این زبان منظم نیست. طبق لم پمپاژ عدد p را پیدا می‌کنیم در ابتدا.
یک رشته‌ی شامل p تا پرانتز باز و p تا پرانتز بسته در نظر بگیرید.
با توجه به این‌که

$$s = xyz,\ |xy| < p, |y| > ۰$$

پس داریم که y شامل تعدادی پرانتز باز است فقط.
حالا با توجه به لم پمپاژ به ازای هر i، یک عضو زبان گفته می‌شود به هر $xy^iz$
اما در این‌جا با قرار دادن i به مقدار ۲، تعداد پرانتزهای باز از بسته بیشتر می‌شود و رشته خوش‌پرانتز نیست.
در نتیجه لم پمپاژ صدق نمی‌کند و زبان منظم نیست.

## ۳.۳

می‌توان با استفاده از زبان $L_۱ = \{a^ib^i \mid i > ۰\}$ نشان داد که زبان $L_۲$ نامنظم است. برای زبان $L_۲$ زبانی را تعریف می‌کنیم که اشتراکش با آن برابر زبان $L_۱$ شود. برای تعریف چنین زبانی می‌توانیم یک زبان که با ۱ شروع می‌شود تعریف کنیم به طوری که حتما ۱ یا ۰ داشته باشد و به تعداد دلخواه تکرار شود، یعنی داریم.

$$L_۴ = \{۱۱^*۰۰^*\} \implies L_۲ \cap L_۴ = L_۱$$

می‌دانیم که زبان $L_۴$ منظم است چون می‌توان یک DFA برای آن طراحی کرد که از دارای سه استیت بوده و با ورود ۱ از استیت شروع به استیت شماره یک می‌رود. در استیت شماره یک با دریافت ورودی صفر به استیت شماره دو و با گرفتن ورودی یک در خود بماند. برای استیت شماره دو هم فقط در حالتی که ورودی، صفر باشد، در خود می‌ماند و حالت $accept$ خواهد بود. حالا چون اشتراک زبان $L_۲$ با زبان دیگری منظم نبود، پس این زبان نامنظم است.

برای بخش دوم این سوال، فرض می‌کنیم $L_۵$ زبانی شامل تمام رشته‌های شامل دقیقا یک عدد ۰ است. این زبان منظم است و عبارت منظم آن به شکل

$$(۱ \cup ۲)^*۰(۱ \cup ۲)^*$$

است. حالا اگر $L_۳$ منظم باشد، با توجه به خواص بستاری زبان $L_۳ \cap L'$ هم منظم است که این زبان معادل زبان $\{۰۱^y۲^y | y \geq ۰\}$

حال چون یک DFA برای این زبان وجود دارد که $\{۱, ۲\} \notin ۰$ پس left quotient این زبان نسبت به $\{۰\}$ می‌شود: $\{۱^y۲^y | y \geq ۰\}$ که زبان نامنظمی است و تناقض داریم و بدست می‌آید که $L_۳$ نامنظم است.