

به نام خدای رنگین کمان

مستند فاز اول پروژه‌ی درس نظریه‌ی زبان‌ها و ماشین‌ها

اعضای گروه:

امیرعلی لطفی

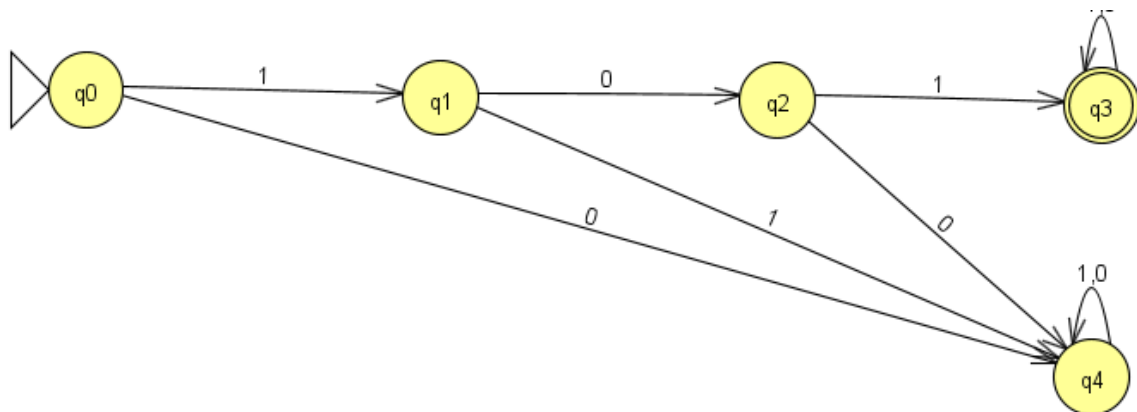
محمد کاظم هرنندی

استاد راهنما : سرکار خانم دکتر اژه‌ای

بهار ۱۴۰۲

زبان‌های منظم - ساخت

DFA



DFA بالا زبان‌هایی به شکل زیر را قبول می‌کند:

$$L(M) = \{ \text{all string with prefix } 101 \}$$

یعنی تمام رشته‌هایی که با ۱۰۱ شروع می‌شوند در این DFA مورد تایید قرار می‌گیرند و رشته‌هایی مانند رشته‌های زیر مورد قبول واقع نمی‌شوند:

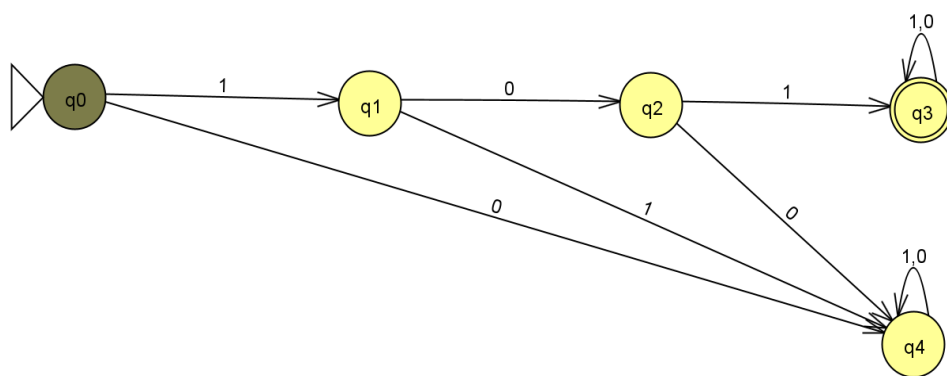
100

001001

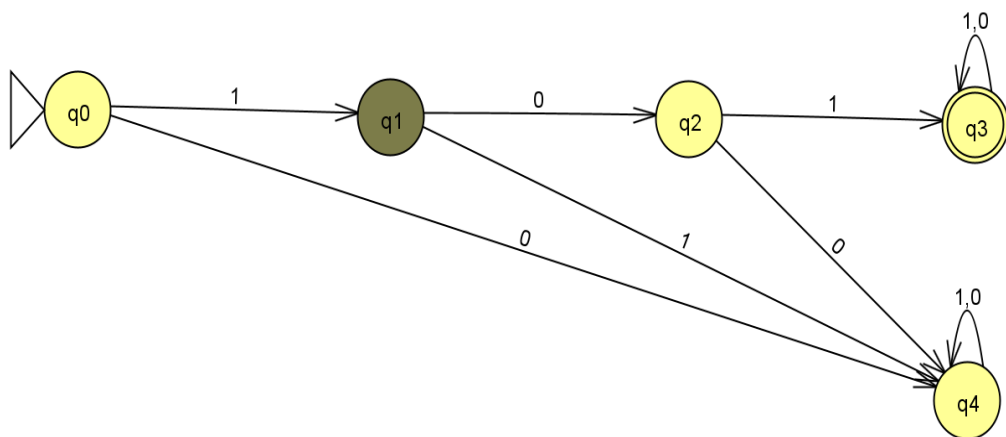
11101

بررسی یک رشته به عنوان نمونه :

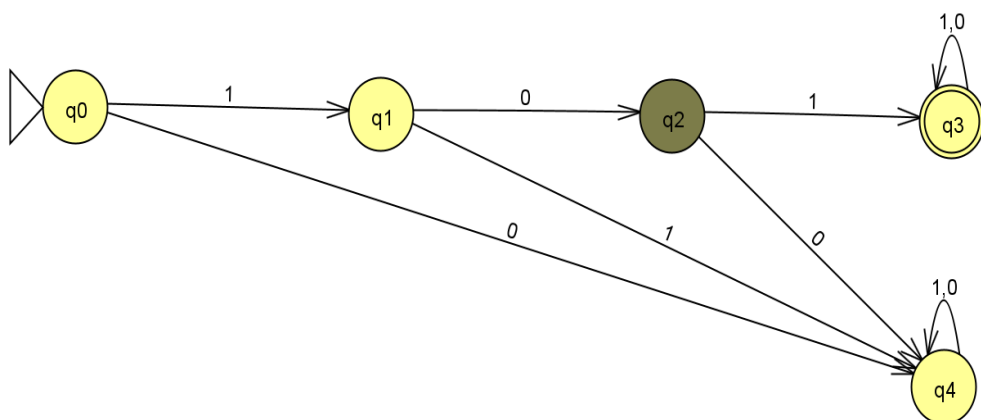
رشته‌ی ۱۰۱۰۰۱ را به DFA می‌دهیم:



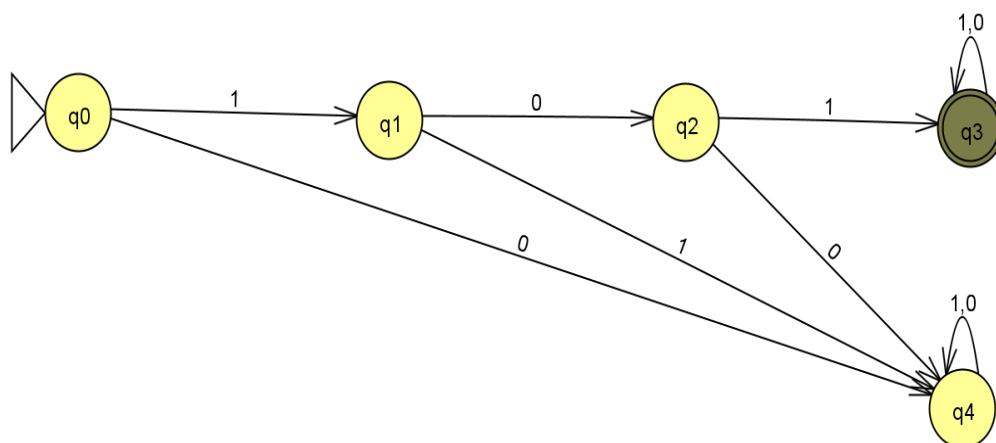
$$q_0 = 101001$$



$$q_1 = 101001$$



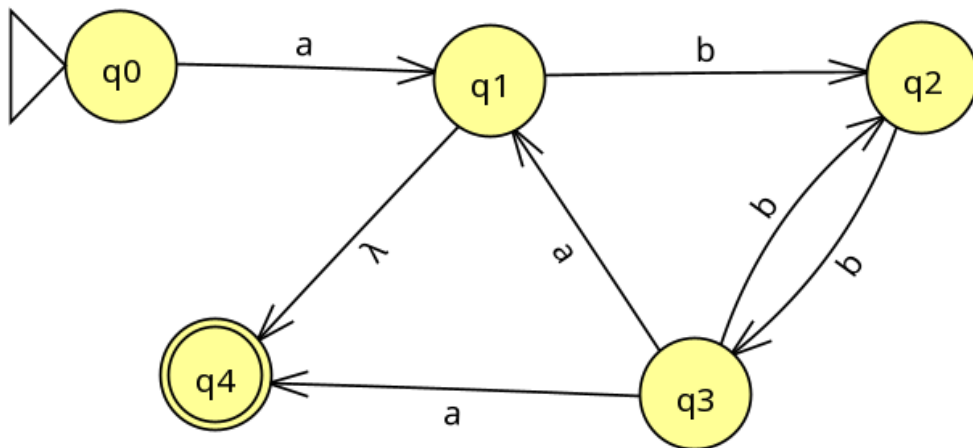
$$q_2 = 101001$$



$$q_3 = 101001$$

و در نهایت این رشته توسط این DFA مورد قبول واقع می‌شود.

ساخت یک NFA



NFA بالا زبانی به شکل زیر را قبول می کند:

$$L = \{a(b^{2n}a)^m \mid n > 0 \wedge m \geq 0\}$$

یعنی رشته‌هایی که با **a** شروع می‌شوند.

سپس به تعداد زوج **b** دارند و یک سمبل **a**. این قسمت می‌تواند به هر تعداد بار تکرار شود.

در آخر هم باید رشته به یک **a** ختم شود.

از جمله رشته‌هایی که در این زبان وجود دارند:

a

abba

abbabbbba

abbbbabbbbabba

و رشته‌هایی که در این زبان وجود ندارند:

aa

abbba

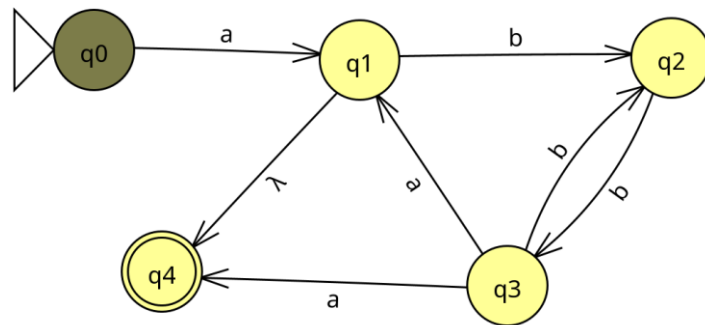
abbabbba

abbaabbbbba

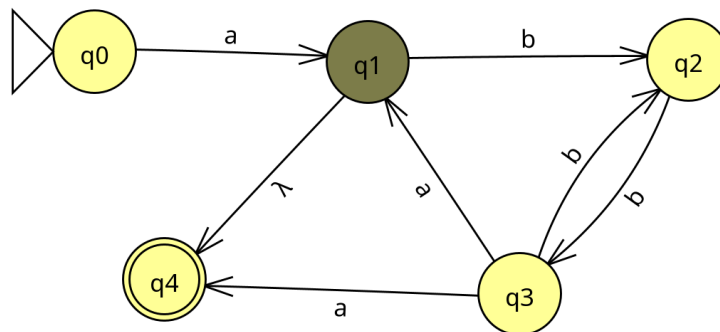
abb

بررسی یک رشته به عنوان نمونه

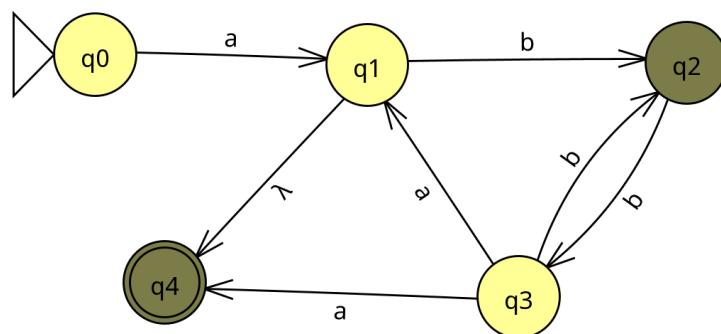
اگر به NFA زیر رشته‌ی **abbabbbba** داده شود، مراحل‌ی که در آن طی می‌شود به صورت زیر است.



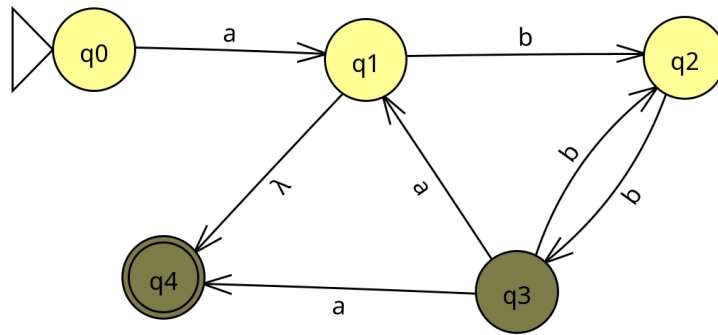
$q_0 = \text{abbabbbba}$



$q_1 = \text{abbabbbba}$



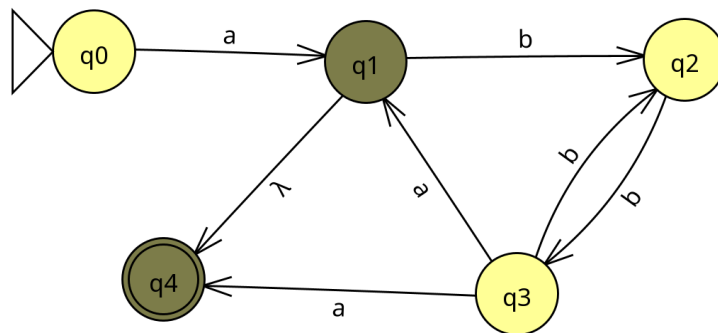
$q_2 = \text{abbabbbba}$



$q_4 = \text{abbabbba}$

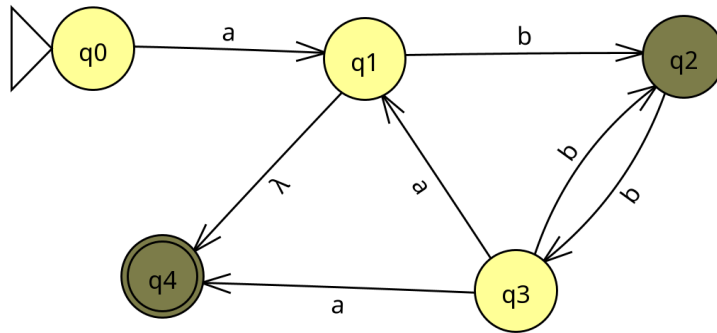
$q_3 = \text{abbabbba}$

$q_4 = \text{abbabbba}$



$q_4 = \text{abbabbba}$

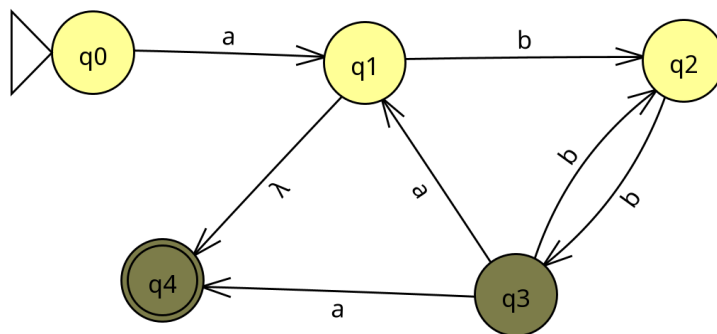
$q_1 = \text{abbabbba}$



$q_4 = abba**bb**ba$

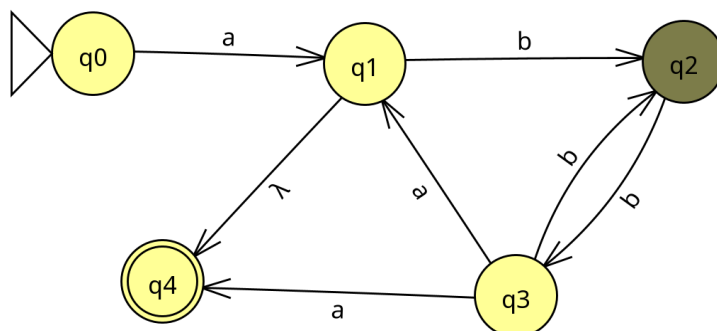
$q_2 = **abb**abbba$

$q_4 = **abb**abbba$

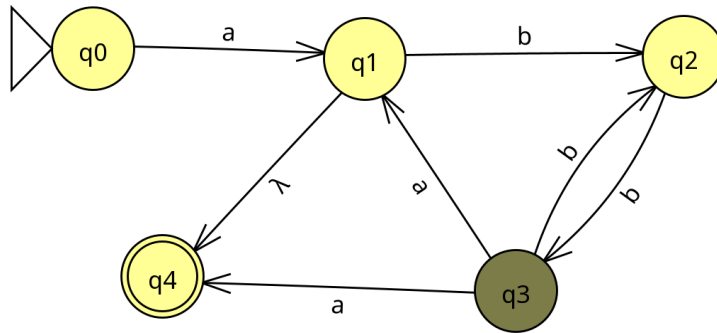


$q_3 = **abb**abbba$

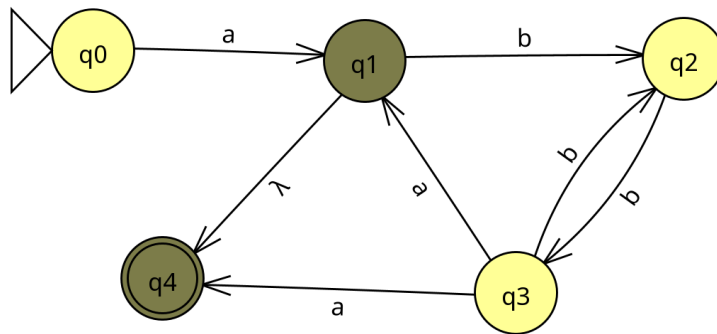
$q_4 = abba**bb**ba$



$q_2 = **abb**abbba$

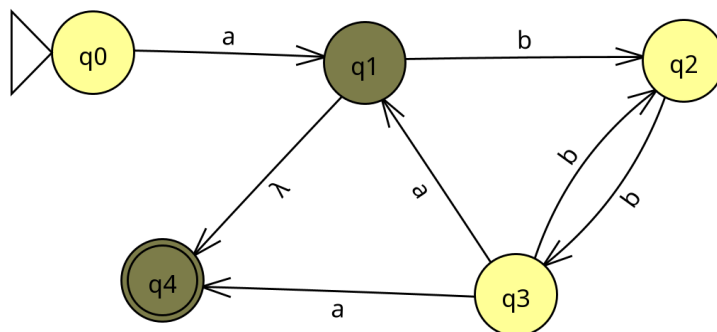


$q_3 = \text{abbabbba}$



$q_4 = \text{abbabbba}$ (accepted)

$q_1 = \text{abbabbba}$

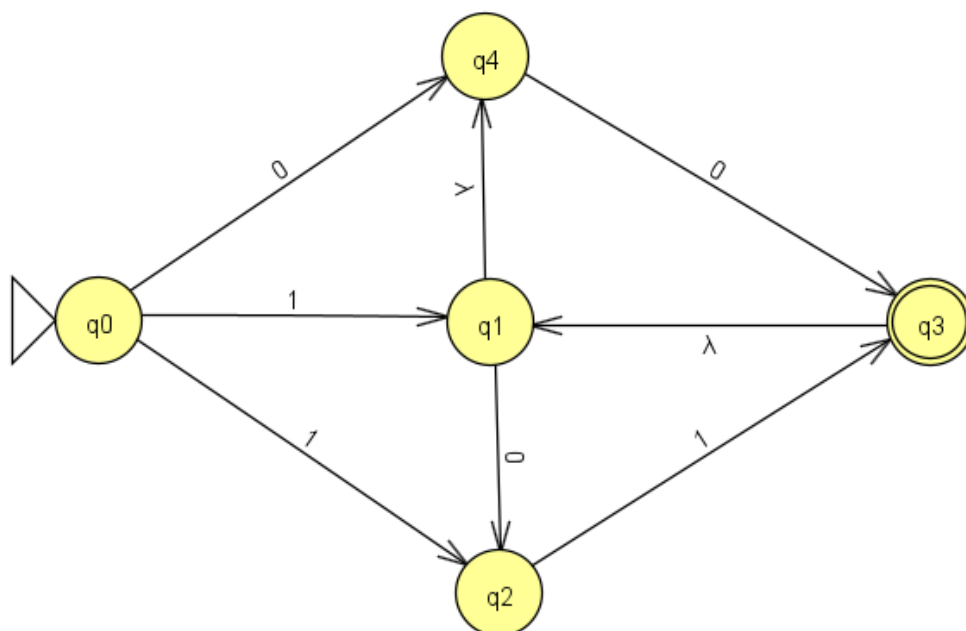


$q_4 = \text{abbabbba}$ (accepted)

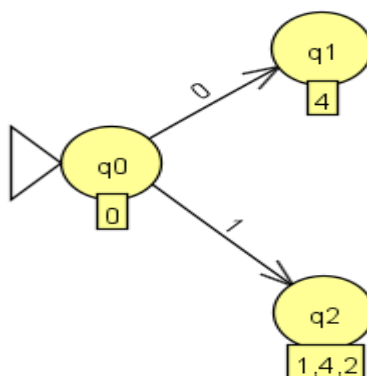
زبان های منظم - تبدیلات

DFA به NFA

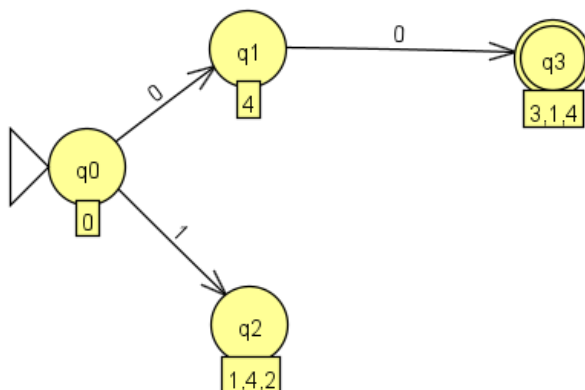
در ابتدا یک NFA رسم می کنیم.



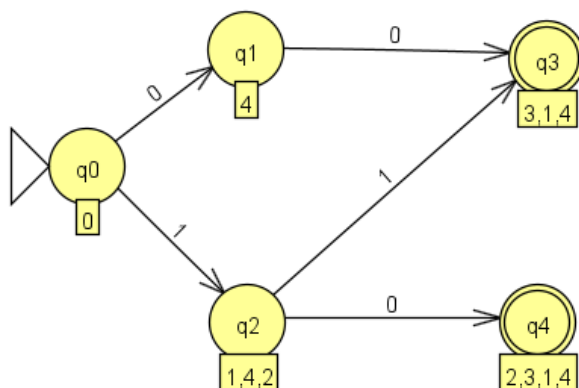
در مرحله اول حالت شروع را مشخص کرده و بعد هر یک از تابع های انتقال حالت آن را رسم می کنیم. سپس به ادامه ی این کار برای همه ی حالت می پردازیم.



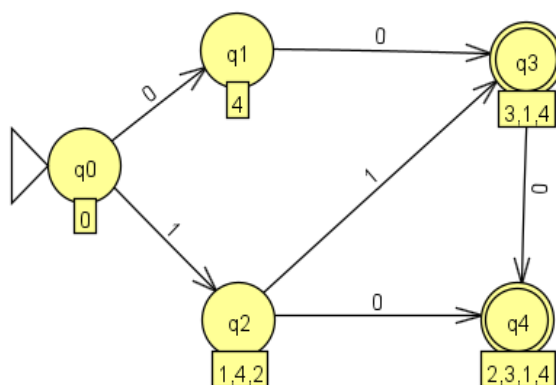
در این مرحله تابع‌های انتقالی از q_1 را رسم می‌کنیم.



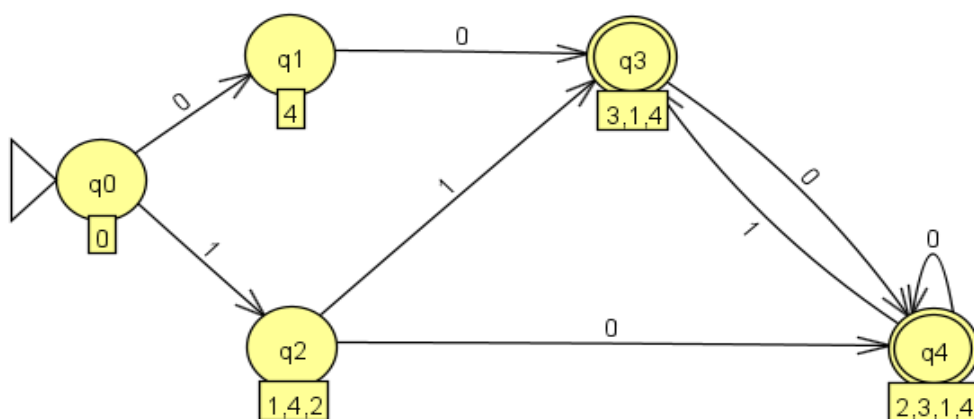
در این مرحله، تابع‌های انتقالی از حالت‌های ۲، ۱ و ۴ NFA را رسم می‌کنیم. برای مثال می‌دانیم که از q_4 با \cdot می‌توانیم به حالت‌های q_1 و q_2 انتقال پیدا کنیم.



در این مرحله انتقال‌های مرحله‌ی q_3 از DFA را کامل می‌کنیم و تنها فقط یک مرحله تا تکمیل باقی مانده است و آن هم انتقال‌های q_4 است.



انتقال های q_4 را رسم کرده و برای بار دیگر DFA خود را مورد بازبینی قرار می دهیم. می توانیم این کار را با نرم افزار هم انجام دهیم.

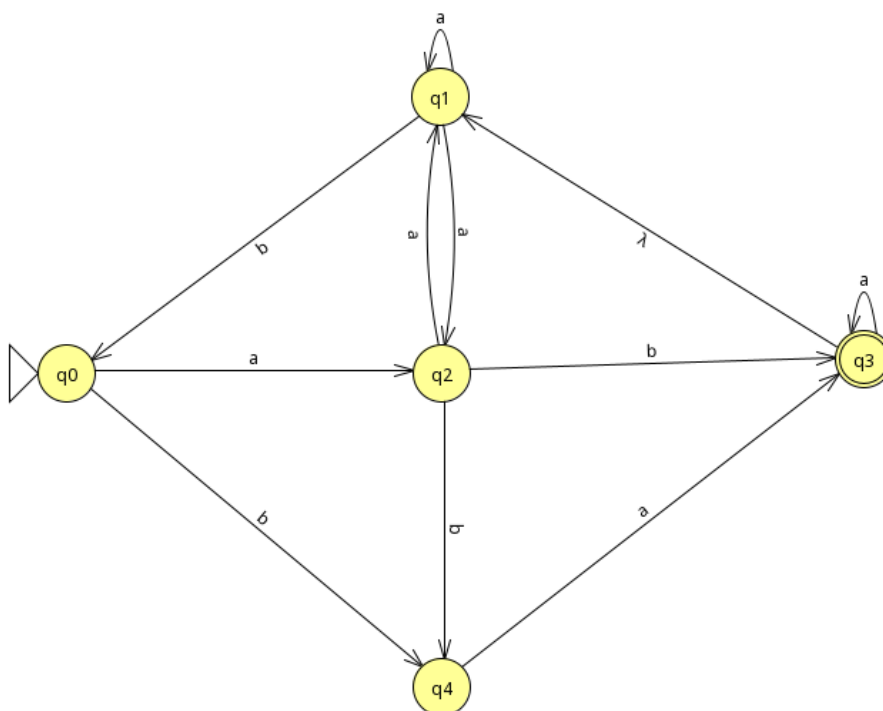


در نتیجه تبدیل NFA به DFA به درستی انجام شده است.

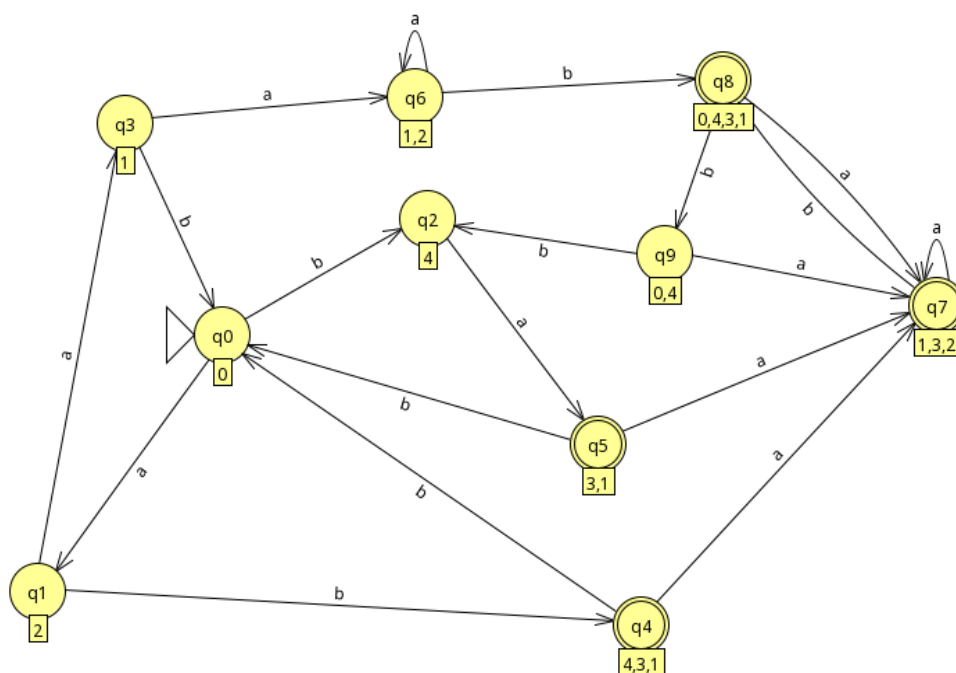


NFA به DFA مینیمم

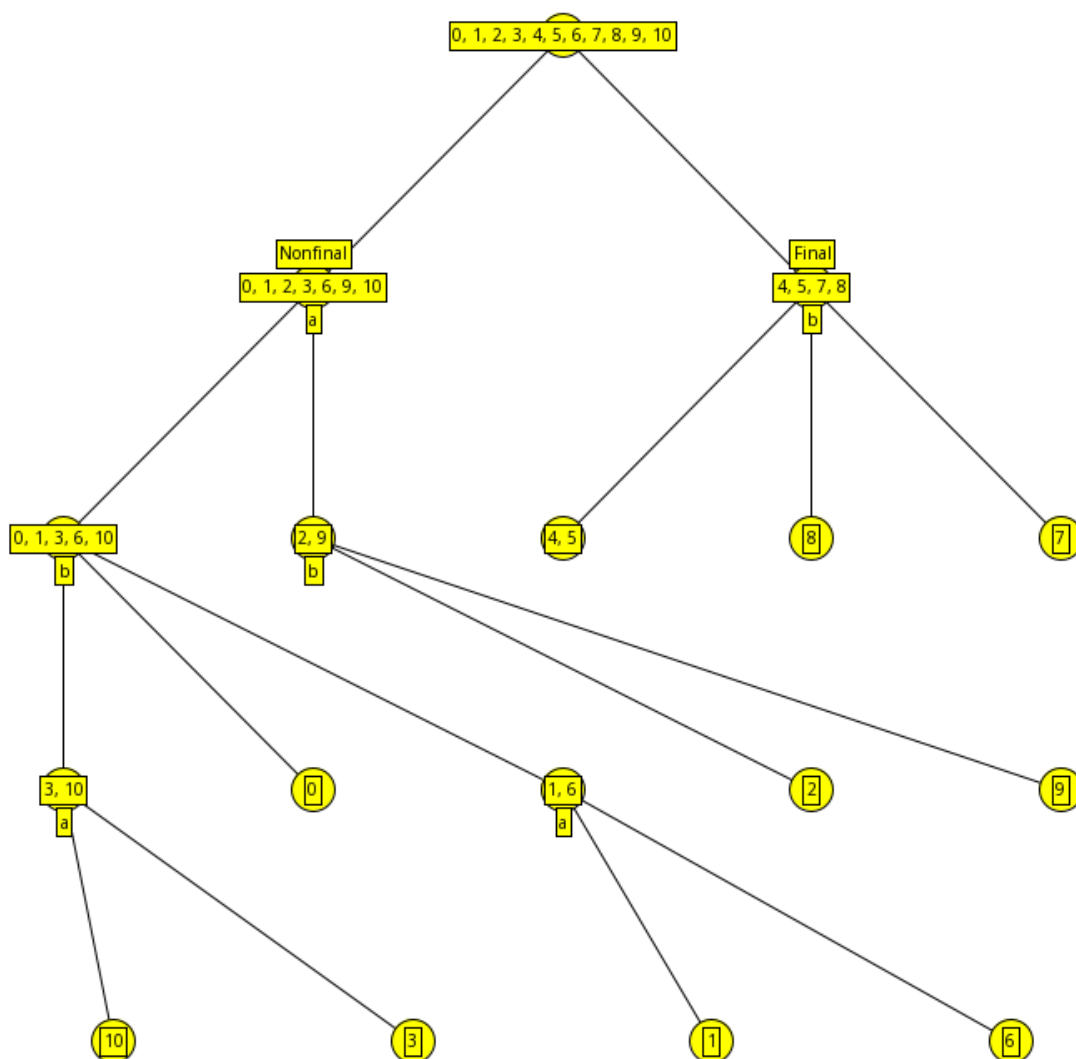
ابتدا یک NFA در نظر می گیریم.



در برنامه JFLAP، با استفاده از گزینه Complete، شکل DFA معادل آن را به دست می آوریم:

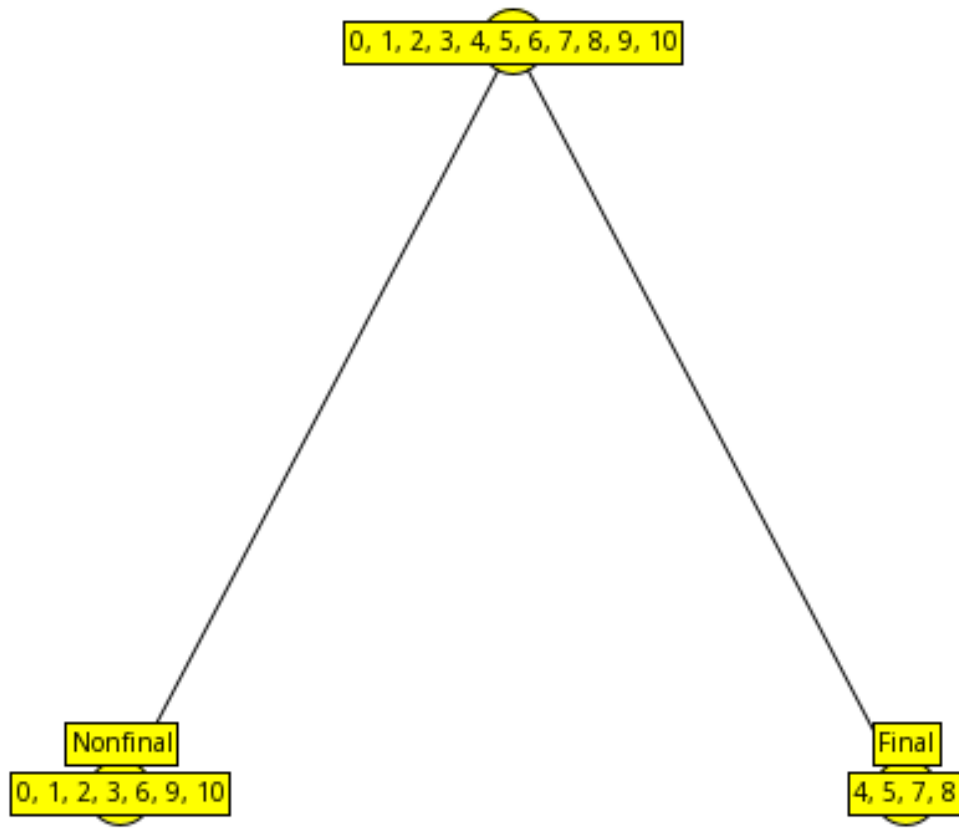


سپس از طریق قسمت Minimize DFA، درخت تقسیم‌بندی حالت‌ها را رسم می‌کنیم:



این درخت به این صورت به وجود می‌آید که ابتدا همه حالت‌ها را به دو قسمت حالت‌های پایانی و حالت‌های غیرپایانی تقسیم می‌کنیم. سپس برای هر گره، به ازای یک تغییر حالت دلخواه، آن دسته از حالت‌هایی که با آن تغییر حالت در همان گره یافت می‌شوند را به یک دسته، و بقیه‌ی حالت‌ها را به دسته‌های جداگانه تقسیم می‌کنیم. این کار را آنقدر ادامه می‌دهیم تا دیگر هیچ گره قابل تقسیمی یافت نشود.

برای درک بهتر چند مرحله ابتدایی این فرایند را تشریح می‌کنیم:

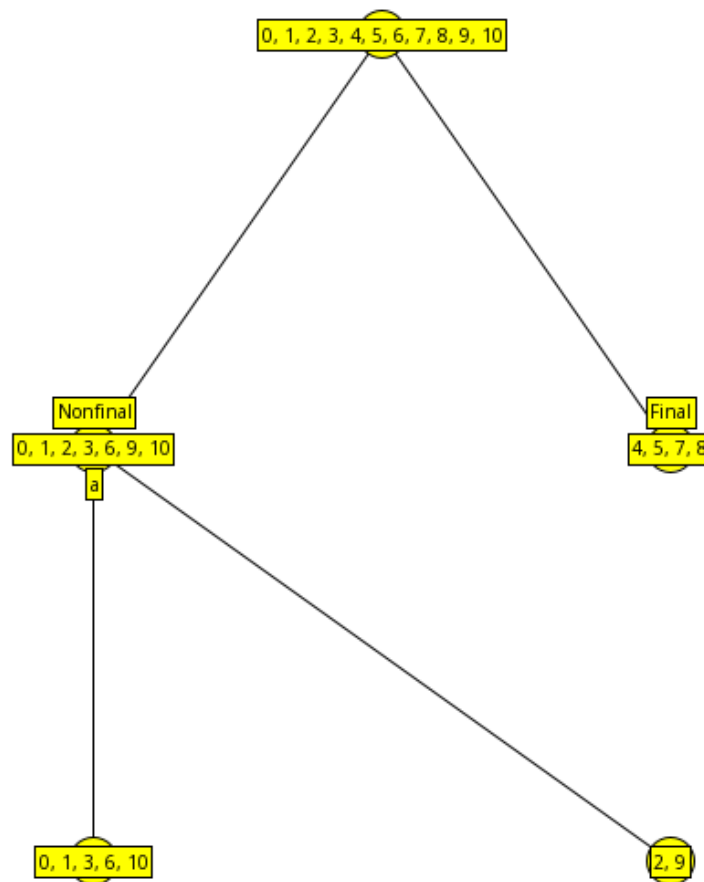


در اینجا تمامی حالات را به دو دسته حالت پایانی (Final) - که شامل q_4, q_5, q_7, q_8 است - و دسته غیرپایانی (Non Final) - که شامل $q_0, q_1, q_2, q_3, q_6, q_9, q_{10}$ است - تقسیم می‌کنیم.

حال دسته Non Final را به ازای سمبل a بررسی می‌کنیم. حالت q_2 به ازای a به q_5 و حالت q_9 به ازای a به q_7 می‌رسد؛ که q_4 و q_9 جز دسته‌ی این گره نیستند.

پس ۲ دسته اضافه می‌کنیم: $۴, ۹$ و $۰, ۱, ۳, ۶, ۱۰$.

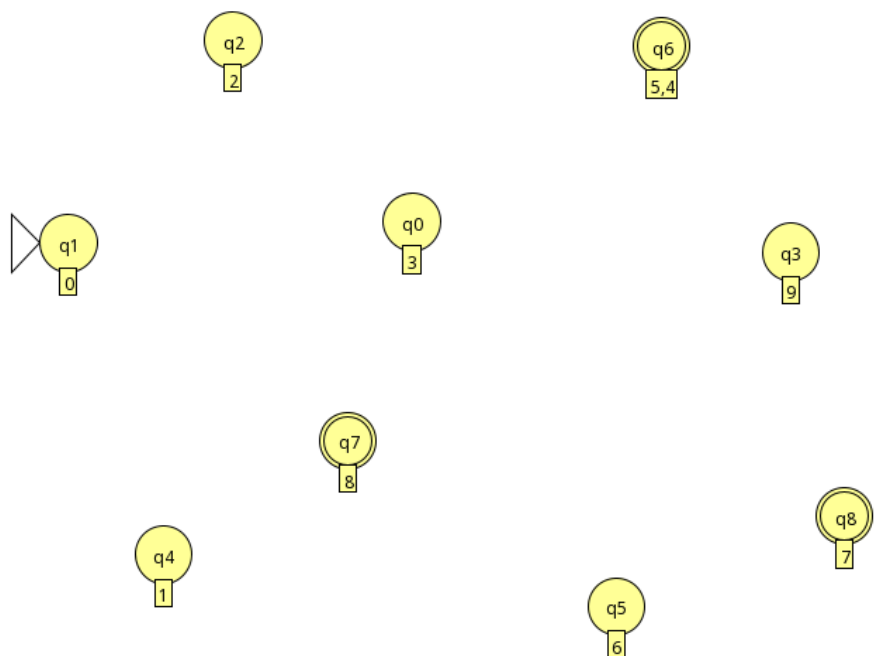
درخت به حالت زیر تبدیل می شود:



سپس همین الگوریتم را برای همه گره ها ادامه می دهیم.

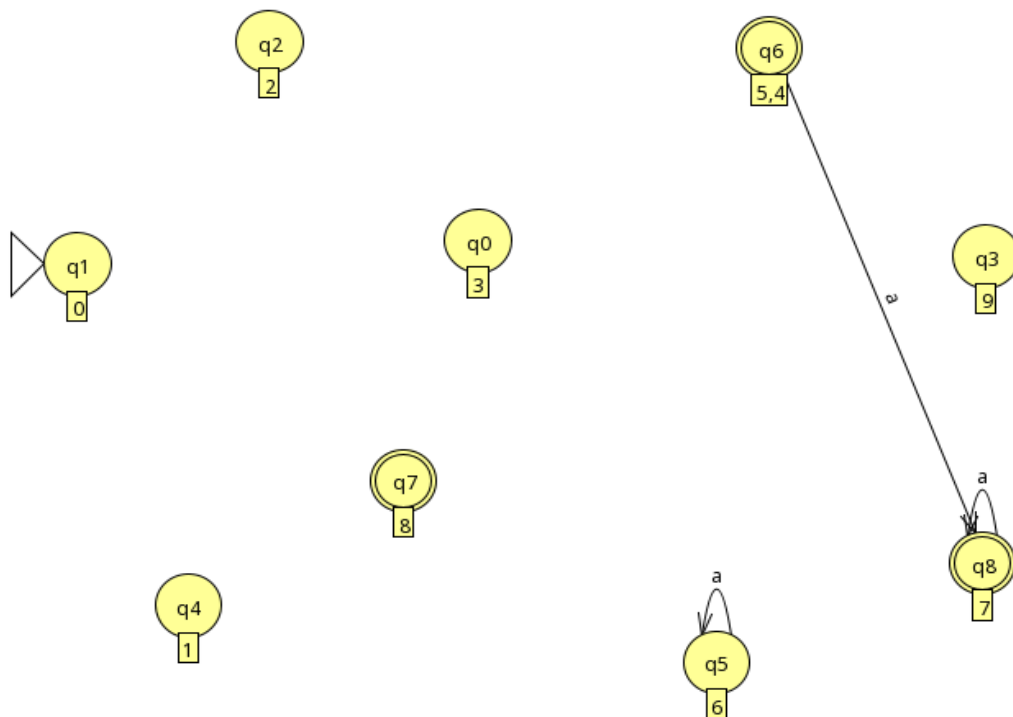
هنگامی که درخت تکمیل شود، دکمه **Finish** نمایانگر می شود. با کلیک بر روی این دکمه به صفحه ای

می رویم که می توان **DFA** مینیوم را تشکیل داد.

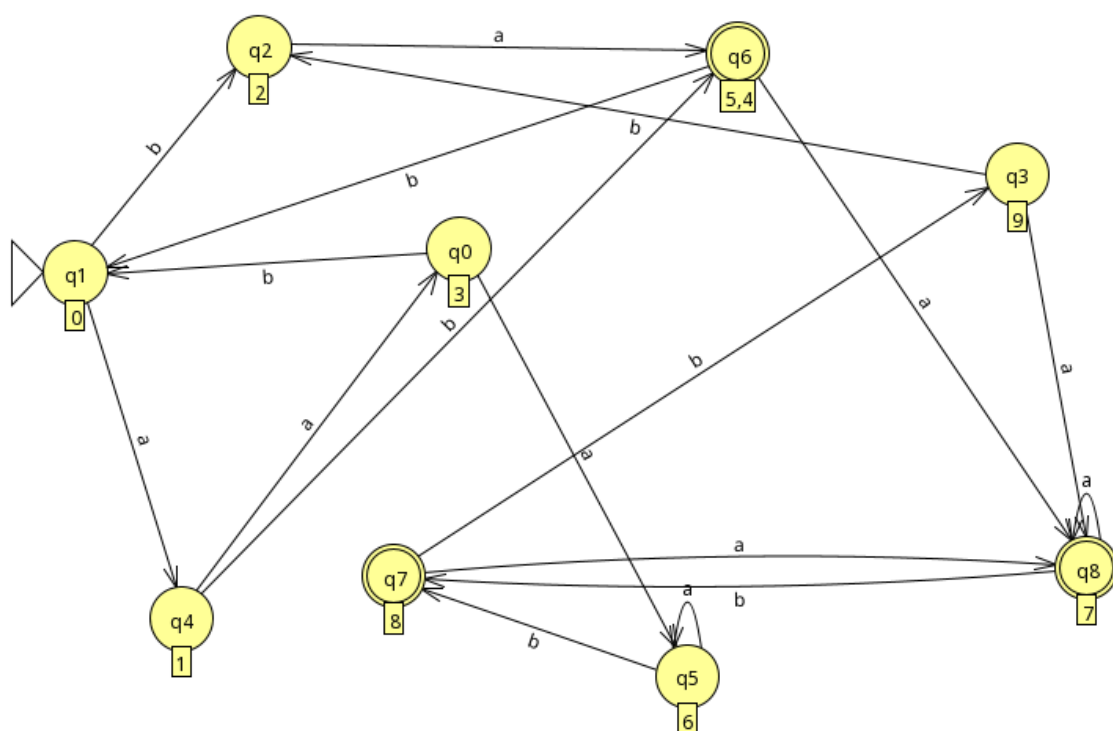


در این مرحله همه‌ی حالت‌ها کشیده شده است و ما باید تغییر حالت‌ها را مشخص کنیم. برای این کار کافی است به ازای هر گره، با توجه به DFA مربوطه، حالت‌ها را به یکدیگر متصل کنیم.

برای مثال حالت‌های ۴ و ۵ هر دو با a به حالت ۷ می‌روند. پس در اینجا q_6 را به q_8 متصل می‌کنیم.



سپس این کار را ادامه می‌دهیم تا DFA مینیموم ما کامل شود.

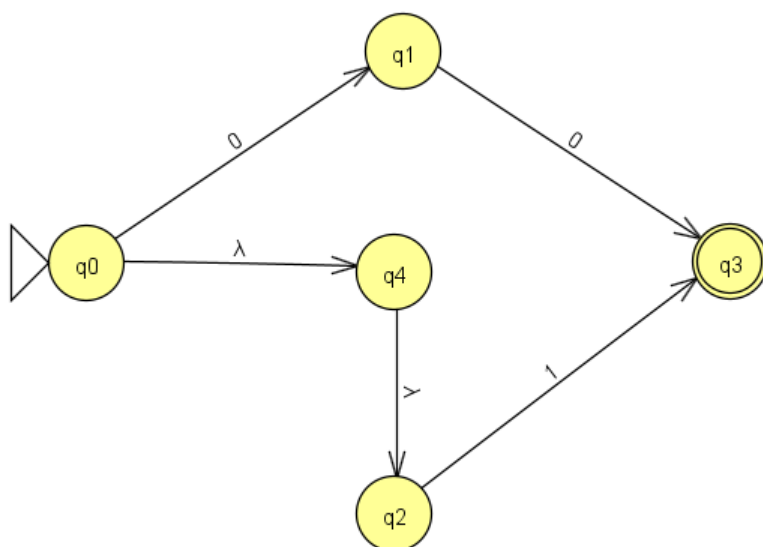


مشاهده می‌شود که DFA ای با ۱۱ حالت به یک DFA با ۹ حالت تبدیل شده است.

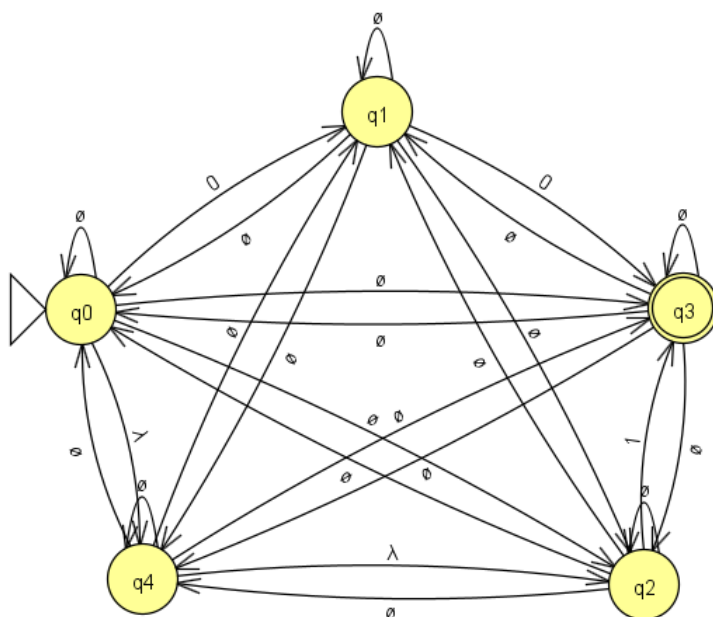
(NFA or DFA) به عبارت منظم

تبدیل یک NFA به عبارت منظم :

در مرحله اول یک NFA ساده رسم می‌کنیم :



بعد از انجام این کار تمام حالت‌های ممکن تابع‌های انتقال را رسم می‌کنیم و با علامت تهی، علامت گذاری می‌کنیم. در حالت کلی می‌توان گفت که این NFA دارای 5^2 تابع انتقال دارد.
(برای خوانا تر بودن گراف، گره حالت q_4 را به پایین شکل انتقال داده‌ایم.)



در مرحله‌ی بعدی باید بتوانیم تمام حالت‌ها به جز شروع و نهایی را حذف کنیم پس باید جدولی از تابع‌های انتقالی که از مرحله‌های حذفی گذر می‌کنند را رسم کنیم. این کار به کمک نرم‌افزار برای حالت‌های مورد نیاز رسم می‌کنیم.

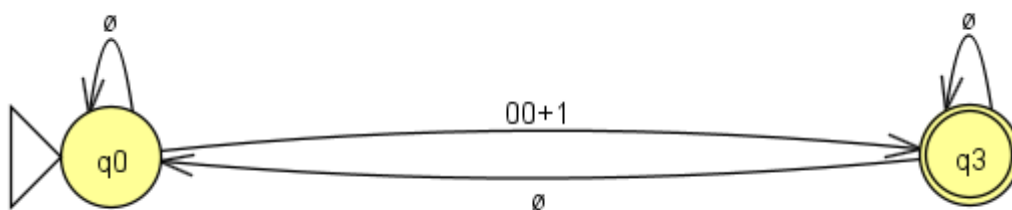
From	To	Label
0	0	\emptyset
0	2	\emptyset
0	3	00
0	4	λ
2	0	\emptyset
2	2	\emptyset
2	3	1
2	4	\emptyset
3	0	\emptyset
3	2	\emptyset
3	3	\emptyset
3	4	\emptyset
4	0	\emptyset
4	2	λ
4	3	\emptyset
4	4	\emptyset

From	To	Label
0	0	\emptyset
0	1	0
0	3	\emptyset
0	4	λ
1	0	\emptyset
1	1	\emptyset
1	3	0
1	4	\emptyset
3	0	\emptyset
3	1	\emptyset
3	3	\emptyset
3	4	\emptyset
4	0	\emptyset
4	1	\emptyset
4	3	1
4	4	\emptyset

From	To	Label
0	0	\emptyset
0	1	0
0	2	λ
0	3	\emptyset
1	0	\emptyset
1	1	\emptyset
1	2	\emptyset
1	3	0
2	0	\emptyset
2	1	\emptyset
2	2	\emptyset
2	3	1
3	0	\emptyset
3	1	\emptyset
3	2	\emptyset
3	3	\emptyset

این جدول‌ها تمامی حالت‌های انتقال را به ما نشان می‌دهد.

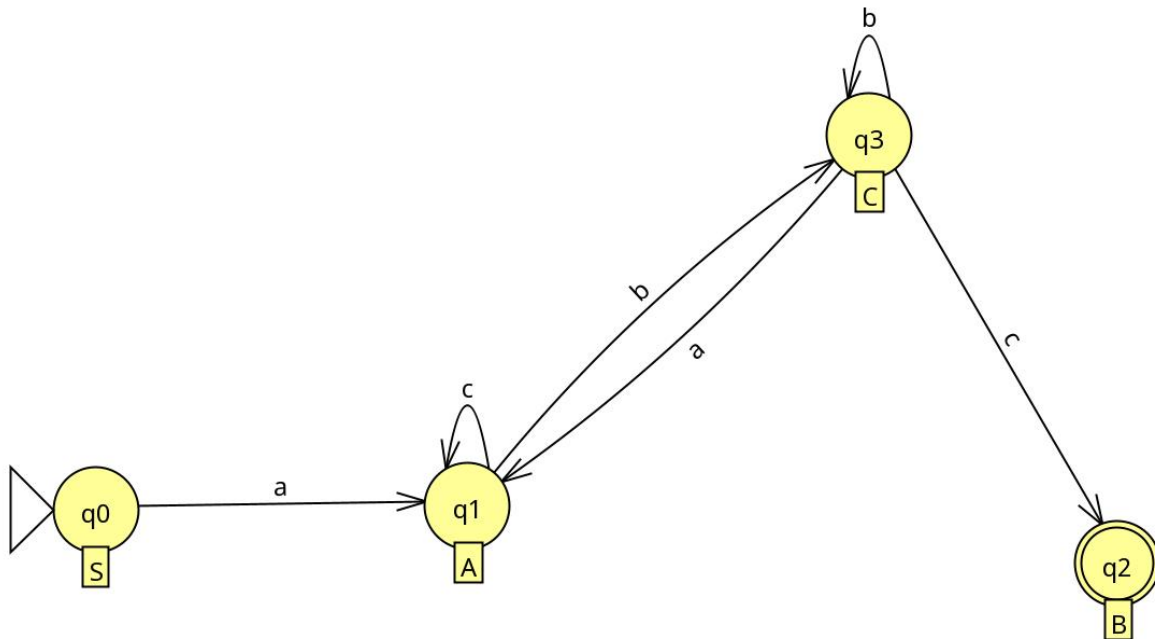
بعد از حذف حالت‌ها تمام تابع‌های انتقال از شروع به پایان را به صورت تابع انتقال در گراف خود رسم می‌کنیم.



بعد از انجام این مرحله کار ما به اتمام می‌رسد و می‌توانیم مشاهده کنیم که عبارت منظم این NFA به صورت $00 + 1$ می‌باشد.

(NFA or DFA) به گرامر منظم

برای این قسمت ابتدا یک DFA دلخواه رسم می کنیم.



برای به دست آوردن گرامر معادل DFA بالا، کافی است به ازای هر تابع تغییر حالت، قانون معادل آن را بنویسیم. به عنوان یک قاعده کلی، برای تابع $\delta(q_0, a) = q_1$ ، معادل قانون گرامر آن $q_0 \rightarrow aq_1$ خواهد بود. همچنین برای هر حالت پایانی q ، قانون $q \rightarrow \lambda$ را خواهیم داشت.

در نرم افزار JFLAP، برای به دست آوردن قانون معادل تغییر حالت از S به A با حرف a ، کافی است روی یال تغییر حالت کلیک کنیم. با این کار قانون $S \rightarrow aA$ به مجموعه قوانین اضافه می شود.

همچنین حالت نهایی q_2 ، بیانگر قانون $B \rightarrow \lambda$ می باشد.

گرامر معادل این DFA به این صورت خواهد بود:

$$G = (\{S, A, B, C\}, \{a, b, c\}, S, P)$$

قوانین P ، به صورت زیر خواهد بود:

LHS		RHS
S	\rightarrow	aA
B	\rightarrow	λ
C	\rightarrow	cB
C	\rightarrow	bC
C	\rightarrow	aA
A	\rightarrow	bC
A	\rightarrow	cA