

نظریه زبان ها و ماشین

دانشگاه آل طه

رشته کامپیوتر

درس تخصصی

پاییز 1404

رئوس مطالب

□ هدف درس:

- معرفی مفاهیم پایه در نظریه محاسبات
- معرفی مدل های صوری مختلف در نظریه محاسبات
- روشن کردن محدودیت های مدل های مختلف محاسباتی

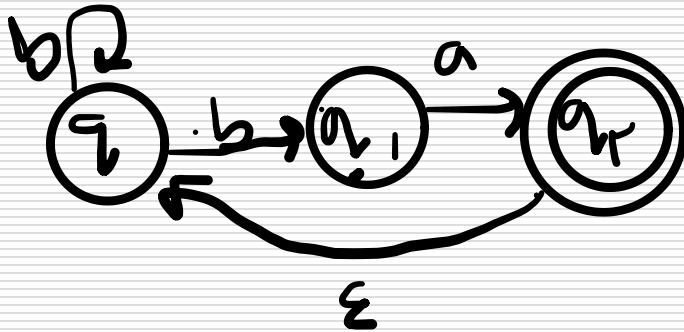
□ سرفصل:

- اتوماتای متناهی ، زبان های منظم و گرامر های منظم
- زبان ها و گرامر های مستقل از متن و اتومانای پشته ای
- زبان ها و گرامر های حساس به متن
- ماشین تورینگ و گرامر نامقید
- تصمیم پذیری
- پیچیدگی محاسباتی

تبدیل nfa به dfa

- نکته: طراحی NFA ساده تر است با این حال کار کردن با DFA راحت تر است.
- سوال: آیا میتوان هر NFA را به DFA تبدیل کرد؟
- دو ماشین وقتی **معادل** هستند که **زبان** هر دو ماشین با هم **مساوی** باشد.
- هدف: تبدیل یک اتوماتای غیرقطعی به قطعی بدون تغییر زبان

تبدیل nfa به dfa



□ نکته: در تبدیل **هر جا** به حالت فاینال رسیدم آن را فاینال در نظر میگیریم.

ایده اصلی

□ در DFA هر حالت دقیقاً یک حالت فعال دارد، اما در NFA ممکن است چند حالت همزمان فعال باشند. بنابراین:

هر حالت DFA = یک مجموعه از حالت‌های NFA

ویژگی	NFA	DFA
تعداد حالت فعال	چندتا	یکی
-ε انتقال	دارد	ندارد
قطعی بودن	✗	✓

□ نکته: در DFA هیچ ابهامی وجود ندارد.

مرحله ۱: حالت شروع DFA

□ اگر ϵ انتقال نداریم: $\text{Start_DFA} = \{q_0\}$

□ اگر ϵ انتقال داریم: $\text{Start_DFA} = \epsilon\text{-closure}(q_0)$

□ تعریف کوتاه $\epsilon\text{-closure}$:

تمام حالت‌هایی که فقط با ϵ قابل دسترسی‌اند.

$$\epsilon - closure(q) = \{q\} \cup \{p \mid \delta^*(q, \epsilon) = p\}$$

مرحله ۲: ساخت انتقال‌ها

□ برای هر حالت DFA و هر نماد ورودی:

■ از تمامی حالت‌های داخل مجموعه حرکت کن

■ نتایج را جمع کن

■ ϵ -closure بگیر

■ $\delta_{\text{DFA}}(S, a) = \epsilon\text{-closure}(\cup \delta_{\text{NFA}}(q, a))$

ساخت جدول DFA

حالت DFA	a	b
{q0}	{q1}	{q0,q2}
{q1}	...	...

مرحله ۳: تکرار

- هر مجموعه‌ی جدید → یک حالت جدید DFA
- این روند ادامه دارد تا حالت جدیدی ساخته نشود
- نکته: همه‌ی حالت‌ها ممکن است استفاده نشوند.

نکات مهم

□ حداکثر حالات 2^n DFA =

□ ϵ -closure فقط در صورت وجود ϵ

□ حالات غیرقابل دسترس حذف می شوند

جمع بندی

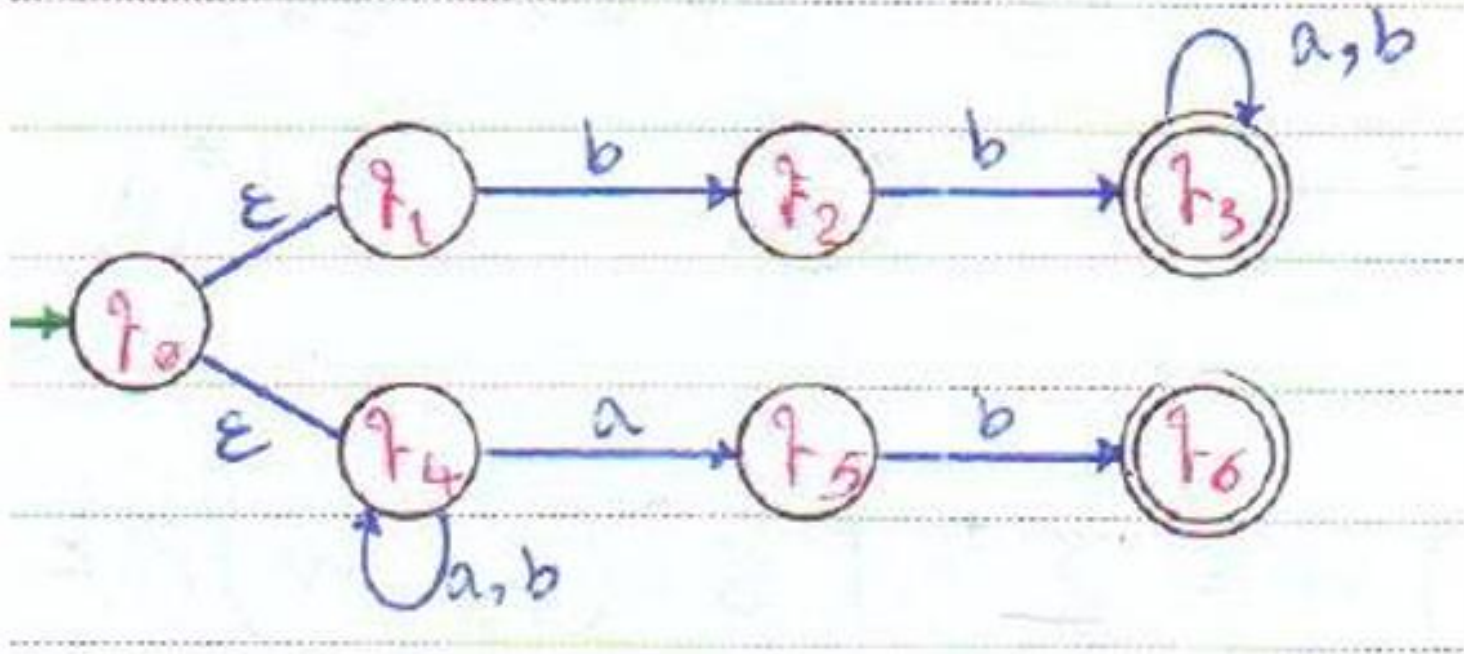
□ مجموعه بساز

□ انتقال محاسبه کن (یه حرف رو انتخاب میکنیم تمام حالت هاشو در نظر میگیریم و اجتماعشو حساب میکنیم)

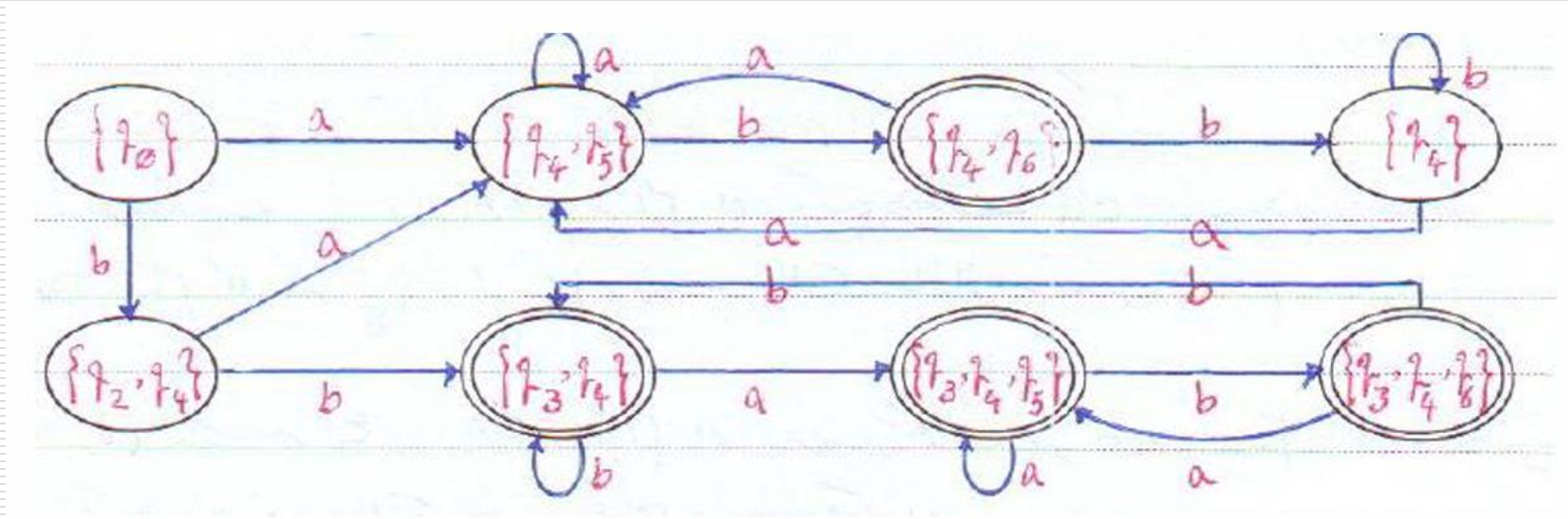
□ تکرار کن

□ پذیرش تعیین کن

مثال



راه حل



تعریف وضعیت‌های ادغام‌پذیر

□ دو یا چند حالت DFA را «ادغام‌پذیر» می‌گوییم اگر:

■ از دید زبان پذیرفته‌شده، هیچ رشته‌ای وجود نداشته باشد که این حالت‌ها را از هم متمایز کند.

■ یعنی هر رشته‌ای که از یکی پذیرفته می‌شود، از دیگری هم پذیرفته می‌شود.

اگر دو حالت: همیشه با هر ورودی رفتار یکسان دارند یا هر دو پذیرش‌اند یا هر دو ناپذیر، پس نگه‌داشتن هر دو اضافه‌کاری است.

تعریف رسمی

$$\square \forall w \in \Sigma^* : \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F$$

□ نکته: وضعیت‌های ادغام‌پذیر بعد از تبدیل NFA به DFA بررسی می‌شوند نه در خود NFA. این بخش مربوط به مینیمم‌سازی DFA است، نه خود تبدیل.

تشخیص وضعیت‌های ادغام‌پذیر (روش مفهومی)

1. همه‌ی حالت‌ها را به دو دسته تقسیم کن:
 - پذیرش
 - غیرپذیرش
2. اگر دو حالت در یک دسته‌اند:
 - انتقال‌هایشان را مقایسه کن
3. اگر با هر نماد ورودی به حالت‌های هم‌دسته می‌روند $\rightarrow$ ادغام‌پذیرند

تعریف وضعیت‌های ادغام‌ناپذیر

□ دو حالت در DFA ادغام‌ناپذیر هستند اگر:

حداقل یک رشته وجود داشته باشد که رفتار این دو حالت را متفاوت کند.

□ یعنی یک ورودی هست که:

■ از یکی به پذیرش برسد

■ از دیگری نرسد

تعریف رسمی

$$\exists w \in \Sigma^* : \delta^*(p, w) \in F \quad \text{و} \quad \delta^*(q, w) \notin F$$

یا بالعکس

□ اگر بتوانی یک رشته پیدا کنی که بگویی: «از این حالت قبول می‌شه، از اون یکی نه» این دو حالت را نباید ادغام کرد.

□ وضعیت‌های ادغام‌ناپذیر یعنی اطلاعات متفاوت زبانی دارند. اگر آن‌ها را یکی کنی زبان تغییر می‌کند DFA دیگر معادل نیست.

روال علامت گذاری

□ روال علامت گذاری جهت تعیین زوج وضعیت های ادغام پذیر و ادغام ناپذیر

1. هر زوج p, q که در آن یکی از آن ها فاینال و دیگری غیر فاینال است را به عنوان زوج وضعیت ادغام ناپذیر مشخص می کنیم (ضربدر می زنیم).

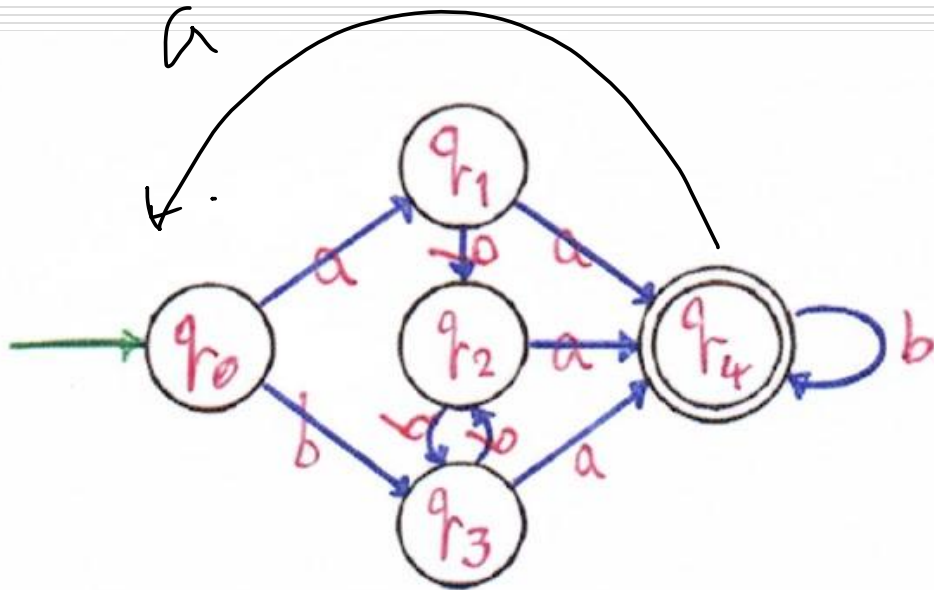
2. گام های زیر را تا مشخص شدن کلیه زوج وضعیت های ادغام ناپذیر تکرار می کنیم:

1. یک زوج وضعیت مثل (p, q) و یک حرف مثل a را انتخاب می کنیم.

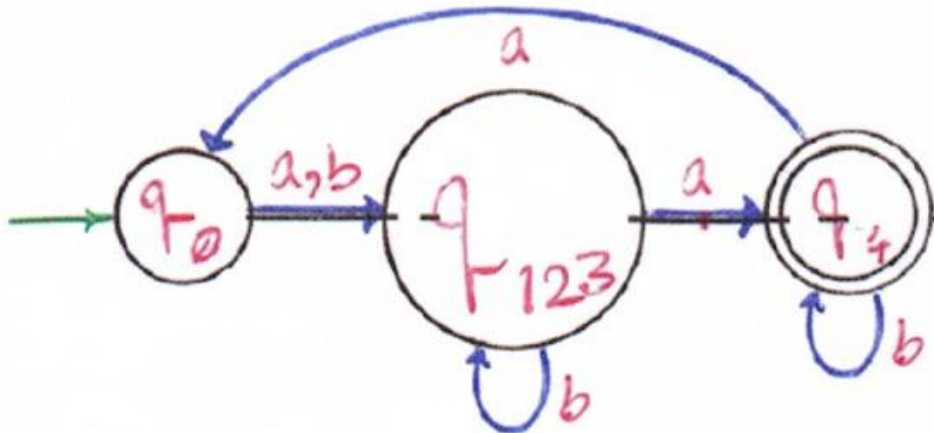
2. $\delta(p, a)$ را محاسبه و آن را s می نامیم و $\delta(q, a)$ را محاسبه و آن را r می نامیم.

3. اگر زوج (s, r) قبلاً به عنوان زوج وضعیت ادغام ناپذیر مشخص شده باشد، زوج وضعیت (p, q) را نیز به عنوان ادغام ناپذیر مشخص می کنیم.

4. کلیه زوج های باقیمانده را به عنوان زوج های ادغام پذیر مشخص کنید.



q_1	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$			
q_2	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$			
q_3	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$			
q_4	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$	$\begin{matrix} \text{r} \\ \text{X} \end{matrix}$
	q_0	q_1	q_2	q_3



روال کاهش dfa

حذف وضعیت‌های غیرقابل دسترس

- مرحله مقدماتی: بیش از شروع فرآیند اصلی، باید تمام وضعیت‌هایی را که از وضعیت شروع (q_0) به آن‌ها هیچ مسیری وجود ندارد، شناسایی و حذف کرد.
- هدف: کوچک‌سازی اولیه ماشین و جلوگیری از محاسبات بیهوده بر روی وضعیت‌های بی‌اثر.

تشخیص وضعیت‌های ادغام‌ناپذیر (همون روال علامت گذاری)

- گام اول (تمایز نهایی): تمام زوج وضعیت‌هایی (p, q) که یکی نهایی و دیگری غیرنهایی است را علامت‌گذاری (X) می‌کنیم.
- گام دوم (تکرار و بررسی): برای زوج‌های باقی‌مانده، با هر حرف الفبا (a) به وضعیت‌های بعدی می‌رویم. اگر وضعیت‌های مقصد ($\delta(p, a), \delta(q, a)$) قبلاً علامت خورده باشند، زوج فعلی هم ادغام‌ناپذیر است و علامت می‌خورد.
- نتیجه: خانه‌هایی که در پایان بدون علامت باقی می‌مانند، وضعیت‌های ادغام‌پذیر (معادل) را نشان می‌دهند.

روال کاهش dfa

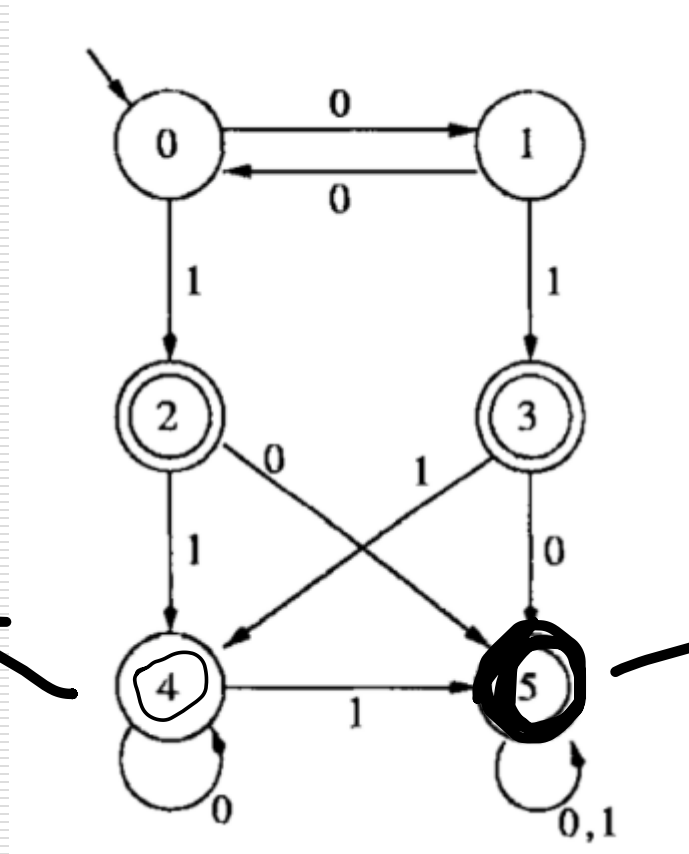
□ ایجاد وضعیت‌های جدید (M') - تمام زوج وضعیت‌هایی که درون یک مجموعه قرار می‌گیرند، ادغام‌پذیر باشند.

- مفهوم افراز (Partitioning): وضعیت‌های معادل را در گروه‌های مجزا قرار می‌دهیم، به طوری که هیچ عضو مشترکی بین گروه‌ها نباشد.
- جمعیت: هر گروه از وضعیت‌های معادل، تبدیل به یک وضعیت واحد در ماشین ساده شده می‌شود.
- مثال: $\{q_i, q_j, q_k\} \rightarrow q_{i...k}$

□ تعیین انتقالات و وضعیت‌های خاص

- رسم یال‌ها: اگر در ماشین اصلی از q_m به q_n می‌رفتیم، در ماشین جدید از وضعیت‌ی که شامل m است به وضعیت‌ی که شامل n است یال رسم می‌کنیم.
- وضعیت شروع: وضعیت‌ی که شامل اندیس صفر ماشین اصلی است، وضعیت شروع جدید خواهد بود.
- وضعیت نهایی: هر وضعیت‌ی که شامل حداقل یک اندیس از وضعیت‌های نهایی ماشین قبلی باشد، به عنوان وضعیت نهایی (M') مشخص می‌شود.

مثال - dfa زیر را کاهش دهید.



$\in P$

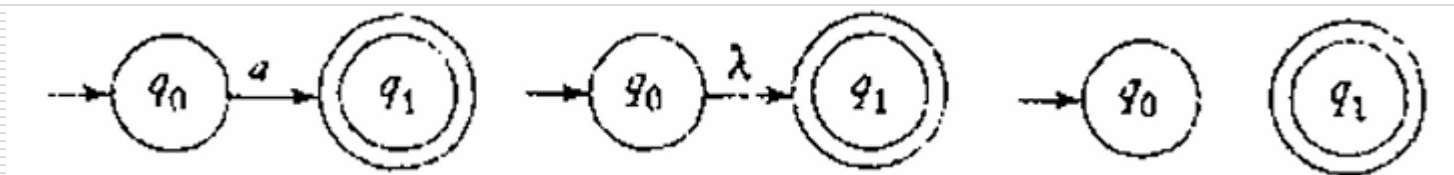


$\notin F$

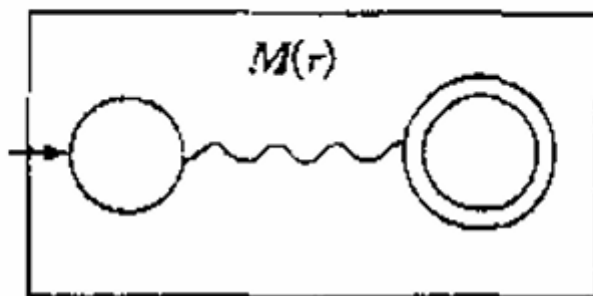


رابطه عبارت منظم و ماشین

- یادآوری عبارت منظم (رجوع شود به اسلاید مربوطه)
- ۱. تهی (ϕ) ماشینی که هیچ رشته‌ای (حتی رشته تهی) را نمی‌پذیرد.
- ۲. رشته تهی (ϵ) ماشینی که بلافاصله و بدون خواندن هیچ حرفی به وضعیت نهایی می‌رود.
- ۳. تک حرف (a) ماشینی که دقیقاً با خواندن حرف a به وضعیت نهایی می‌رسد.

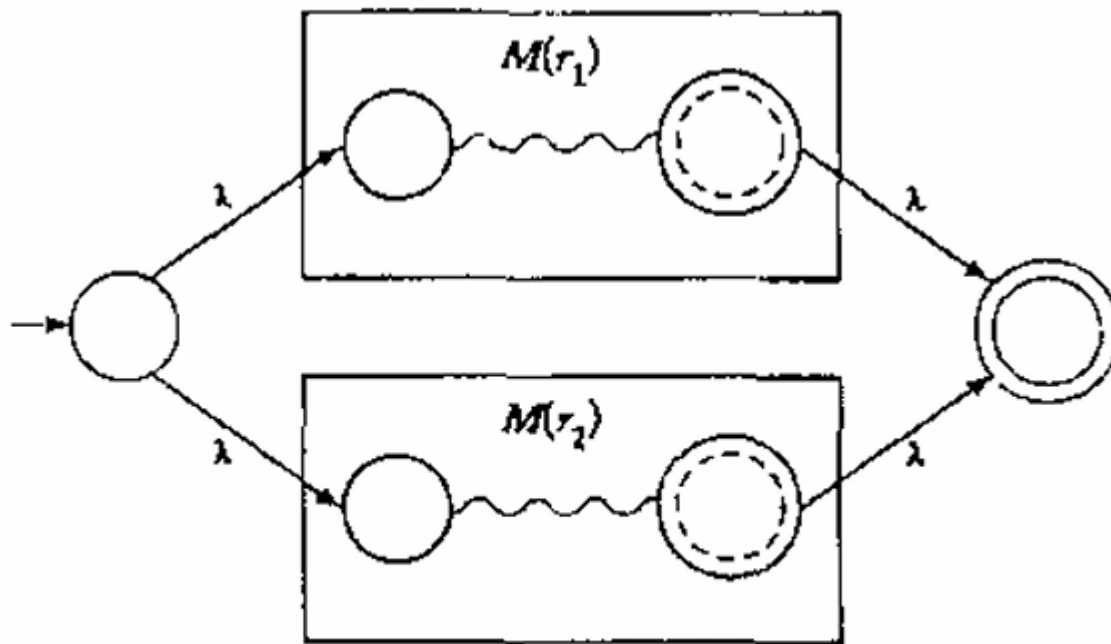


رابطه عبارت منظم و ماشین



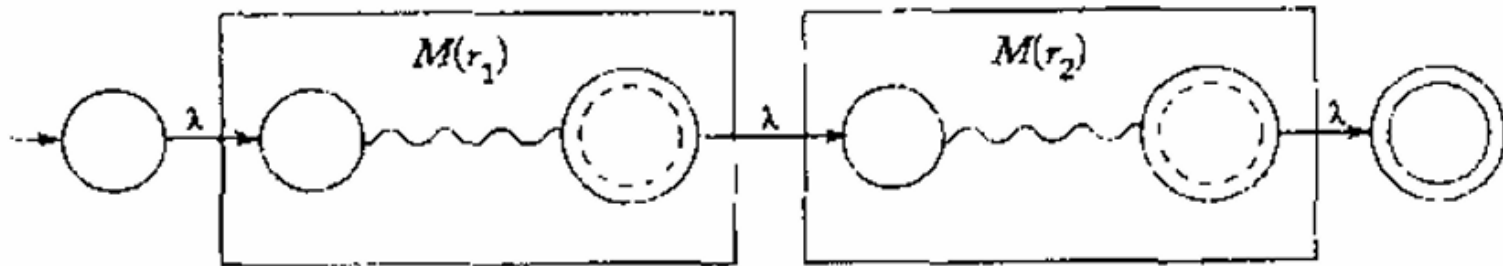
شکل ۳-۲ نمایش طرح کلی یک پذیرنده متناهی غیرقطعی که $L(r)$ را می پذیرد.

رابطه عبارت منظم و ماشین



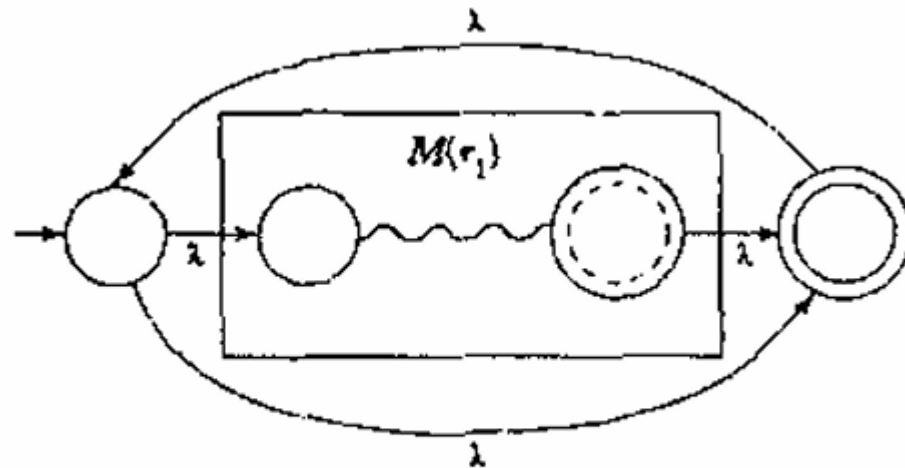
شکل ۳-۳ ماشین برای $L(r_1 + r_2)$.

رابطه عبارت منظم و ماشین



شکل ۳-۴ ماشین برای $L(\eta r_2)$.

رابطه عبارت منظم و ماشین



شکل ۳-۵ ماشین برای $L(r_1^*)$.

تبدیل عبارت منظم به nfa

□ ماشینی که عبارت زیر را می پذیرد.

$$L((a + bb)^*(ba^* + \lambda)) \quad \square$$

تبدیل عبارت منظم به nfa

$$L((a + bb)^*(ba^* + \lambda)) \quad \square$$

