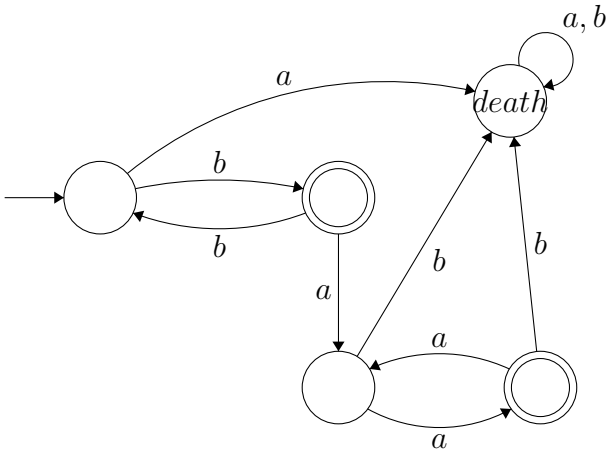
حل تکلیف سری دوم

مبانی نظریه محاسبه. دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. ترم ۴۰۳۲

۱. $b^+(bb)^*(aa)^*$



۲. اگر بخواهیم پروسه بهینه سازی dfa را روی این ماشین اعمال کنیم، در مرحله اول زوج وضعیتهایی که کاندید برای اغام شدن هستند را می‌نویسیم.

$$\{(a,b),(b,f),(a,f),(c,d),(d,e),(c,e)\}$$

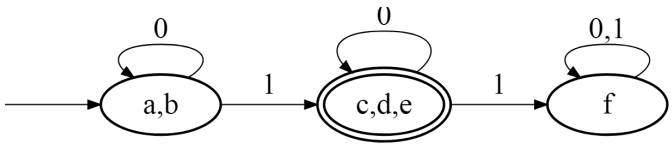
چون a و b با 1 به یک وضعیت پذیرش می‌روند ولی f به (خودش) غیر پذیرش می‌رود پس زوجهای (a,f) و (b,f) قابل ادغام نیستند و حذف می‌شوند.

$$\{(a,b),(c,d),(d,e),(c,e)\}$$

با در نظر گرفتن ورودیهای مختلف، در این مرحله هیچ دو وضعیتی حذف نمی‌شوند و لذا سه کلاس از وضعیتها ایجاد می‌شود.

$$q_1=\{a,b\}, \qquad q_2=\{c,d,e\}, \qquad q_3=\{f\}$$

در نهایت ماشین بصورت زیر خلاصه می‌شود که معادل با رشته‌هایی است که فقط یک 1 دارند.



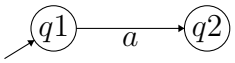
۳. در مورد زبان B، رشته $w = c^p a^p b$ تناقض مورد نظر را ایجاد می‌کند. چون y باید از p کاراکتر اول انتخاب شود، لذا تزریق آن باعث خارج شدن رشته از زبان می‌شود.

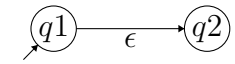
در مورد زبان C رشته $w = a^p b a^p b a^p b$ می‌تواند تناقض مورد نظر را ایجاد کند. مانند مورد بالا قسمت y باید از p کاراکتر اول رشته انتخاب شود که تزریق آن باعث خروج رشته از زبان می‌شود.

۴. ابتدا نشان می‌دهیم که هر زبان منظم توسط یک all-nfa پذیرفته می‌شود. این یک کار آسان است چون dfa در واقع نوعی از all-nfa است و چون هر زبان منظم یک dfa دارد پس گزاره درست است.

از طرف دیگر، نشان می‌دهیم زبانی که توسط یک all-nfa پذیرفته می‌شود منظم است. برای اثبات این گزاره، ادعا می‌کنیم یک all-nfa را می‌توان به dfa تبدیل کرد. انجام این کار مانند تبدیل nfa به dfa است با این تفاوت که وقتی وضعیتهای پذیرش را مشخص می‌کنیم، فقط زیرمجموعه‌هایی را پذیرش قرار می‌دهیم که بطور خالص در ماشین اولیه وضعیت پذیرش باشند.

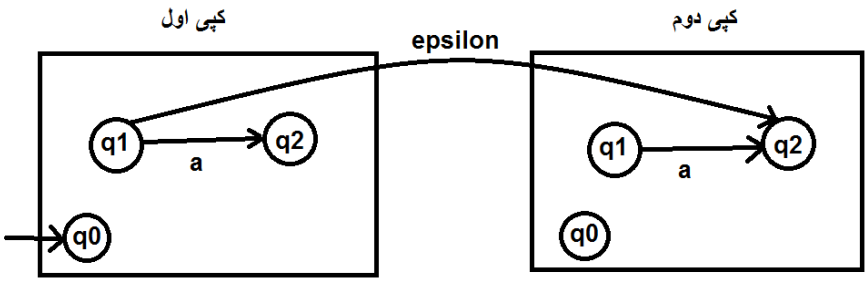
۵. چون A منظم است پس یک ماشین معین dfa دارد. فرض کنید اسم این ماشین M باشد. یک فلش با برچسب a را در نظر بگیرید. با تبدیل برچسب فلش به ϵ مثل این است که یک کاراکتر از یک رشته مورد قبول حذف شده.



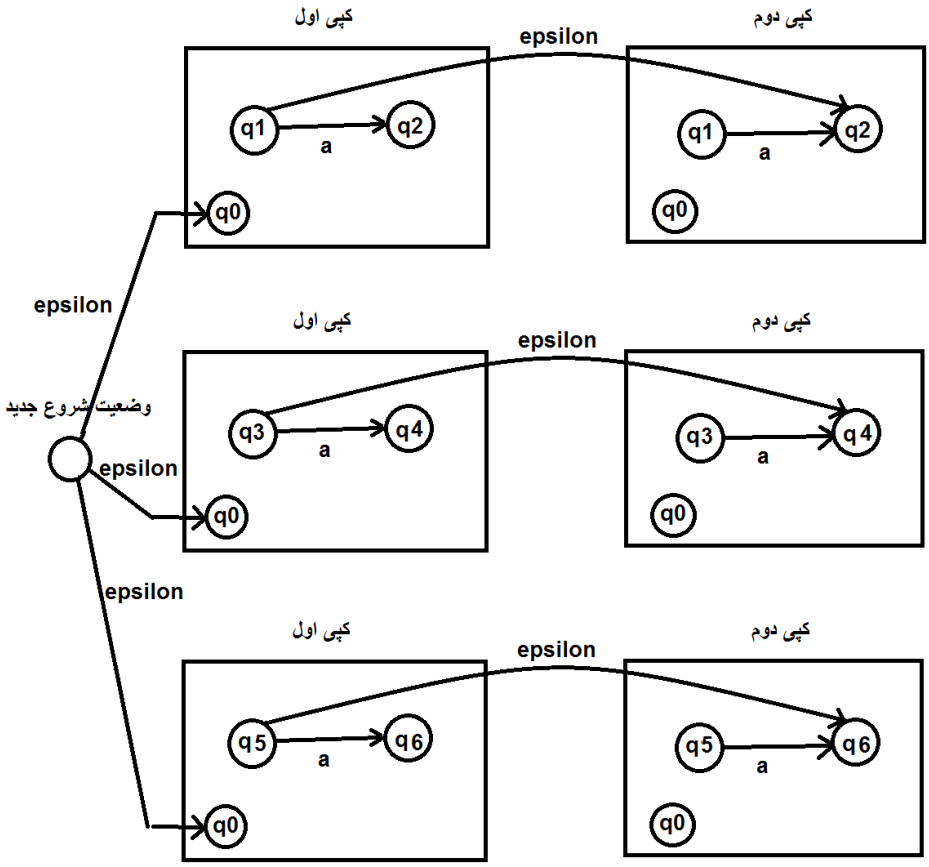


اما چند مشکل وجود دارد. برچسب کدام فلش را ϵ کنیم (باید همه فلش ها را در نظر بگیریم). دوم و مهمتر اینکه ممکن است که فلش مورد نظر در یک دور قرار گرفته باشد و لذا ϵ کردن برچسب فلش می‌تواند باعث حذف چندین کاراکتر از رشته شود.

برای حل چالش دوم، دو کپی از ماشین را ایجاد می‌کنیم و یک فلش ϵ از q_1 در کپی اول به وضعیت q_2 در کپی دوم قرار می‌دهیم. بدین ترتیب فلش ϵ ایجاد شده فقط یک بار استفاده می‌شود و فقط یک کاراکتر از رشته حذف می‌شود. همینطور باید وضعیت پذیرش در کپی اول را به غیر پذیرش تغییر دهیم تا ماشین را مجبور کنیم که حداقل یک بار از فلش ϵ استفاده کند. درضمن، وضعیت شروع در کپی دوم هم نباید وضعیت شروع باشد.



این ساختار را برای همه فلشهای ماشین انجام می‌دهیم. پس به تعداد $2m$ کپی از ماشین اولیه نیاز داریم وقتی که m تعداد فلشها در ماشین اولیه باشد. چون m متناهی است، تعداد وضعیتهای ماشین ایجاد شده نیز متناهی است. یک مثال در صفحه بعد نشان داده شده است.


کپی اول
کپی دوم
epsilon
q1
a
q2
q0
epsilon
وضعیت شروع جدید
epsilon
epsilon
q3
q4
q5
q6