



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

## پروژه درس نظریه زبان ها و ماشین ها

استاد درس: دکتر احمد رضا منتظرالقائم

دستیاران تدریس:

فائزه صالحی

نسرين اسحاقیان

حسین حسینی

محمد حسین رنگرز

فاطمه سادات شجاعی

محمد حسین ملکی

1403-1402

در این پروژه قصد داریم توصیفی از یک NFA را از ورودی بخوانیم و سپس آن را به یک DFA معادل تبدیل کنیم. در نهایت توصیفی از DFA را به شکلی که توضیح داده خواهد شد چاپ کنیم.

هر DFA یا NFA را میتوان توسط پنج مولفه توصیف کرد:

1. الفبا
2. مجموعه حالات
3. حالت شروع
4. حالت(های) پذیرش
5. تابع انتقال

فرض می‌کنیم الفبا متشکل از دو نماد باشد.

$$\{a, b\}$$

همچنین مجموعه حالات را با اعداد صحیح با شروع از صفر برچسب‌گذاری می‌کنیم و در مورد NFA همیشه حالتی که برچسب صفر دارد را حالت شروع در نظر می‌گیریم. در نتیجه برای توصیف یک NFA تنها نیاز به دانستن تعداد حالتها، حالت(های) پذیرش و تابع انتقال خواهیم داشت اما در مورد DFA علاوه بر اینها باید برچسب حالت شروع نیز مشخص شود. همچنین در نظر داشته باشید که برای نمایش اِپسیلون از نماد «'» استفاده می‌کنیم.

### برچسب‌گذاری حالت‌های DFA

مثلا اگر یک NFA با سه حالت داشته باشیم؛ طبق قاعده‌ای که بالاتر بیان شد، برچسب حالت‌های آن 0، 1 و 2 خواهد بود. در مورد DFA نیز برچسب حالت‌ها از 0 تا 7 خواهد بود (الزاما نیاز به استفاده از همه برچسب‌ها نخواهیم داشت) که به شکل زیر متناظر با حالت‌های NFA می‌شوند.

برچسب حالت‌های DFA	حالت‌های متناظر در NFA
0	{}
1	{0}
2	{1}
3	{2}
4	{0, 1}
5	{0, 2}
6	{1, 2}
7	{0, 1, 2}

حال برنامه‌ای بنویسید که از ورودی تعداد حالت‌ها، حالت‌های پذیرش و تابع انتقال یک NFA را دریافت کند و در خروجی همین پارامترها به علاوه حالت شروع را برای DFA چاپ کند (نیازی به کمینه‌سازی DFA نیست اما با این وجود الزاماً نیاز به استفاده از همه برچسب‌ها نخواهیم داشت).

## ورودی

در خط اول ورودی تعداد حالت‌های NFA قرار دارد. در خط دوم نیز برچسب حالت(ها) پذیرش به شکل مرتب‌شده (صعودی) قرار می‌گیرند که بدون فاصله و با استفاده از «» از یکدیگر جدا می‌شوند. در هر کدام از خطوط بعدی نیز یک بخش از تابع انتقال به شکل

<from state>:<symbol>:<to state1>,<to state2>,. . .

<from state>:<symbol>:<to state1>,<to state2>,. . .

. . .

است (تابع انتقال به صورت کاملاً مرتب‌شده داده می‌شود).

- در خط آخر ورودی جهت مشخص شدن پایان ورودی و راحت‌تر خواندن آن کلمه «end» قرار گرفته است.

## خروجی

در خط اول خروجی باید تعداد حالت‌های DFA چاپ شود و در خط دوم برچسب حالت شروع. در خط سوم نیز باید حالت(های) پذیرش DFA به صورت مرتب (صعودی) و با «» از یکدیگر جدا شده چاپ شوند. در هر یک از خطوط بعدی نیز باید به شکل زیر تابع انتقال به صورت مرتب‌شده و با ترتیب Lexicographic چاپ شود.

<from state>:<symbol>:<to state>

<from state>:<symbol>:<to state>

. . .

## مثال

در اینجا نمونه‌ای برای فهم بهتر صورت سوال و قالب ورودی و خروجی داده می‌شود.

### ورودی نمونه

این مثال مربوط به زبان  $a^*$  است.

```
1
0
0:a:0
end
```

### خروجی نمونه

```
2
1
1
0:a:0
0:b:0
1:a:1
1:b:0
```

## نکات تحویل

- پروژه باید به صورت فردی انجام شود و حل گروهی آن مجاز نیست. همچنین در صورت کشف تقلب نمره 100- لحاظ می شود.
- محدودیت زبان برنامه نویسی برای پروژه وجود ندارد و مجاز به استفاده از هر زبانی هستید.
- فایل کد باید تا زمان تعیین شده در کوئرا آپلود شود.
- مهلت ارسال پروژه: 14 فروردین