به نام خدا

گزارش کار پروژه:
پردازش گرامرهای منظم و تبدیل آن‌ها به ماشین‌های متناهی

در پروژه هدف اصلی این بود که گرامرهای منظم را پردازش کنیم، آن‌ها را به NFA تبدیل کنیم، سپس با استفاده از الگوریتم مجموعه حالات آن‌ها را به DFA تبدیل کرده و در نهایت عملیات‌هایی مانند اجتماع، اشتراک و متمم را روی آن‌ها اعمال کنیم.

طراحی کلاس‌ها و ساختار کد :

برای پیاده‌سازی پروژه، ابتدا به یک ساختار واضح و سازمان‌یافته نیاز داشتم. بنابراین سه کلاس اصلی ایجاد کردم:

- کلاس RegularGrammar برای ذخیره و پردازش گرامرهای ورودی.
- کلاس Parser برای خواندن فایل و تجزیه اطلاعات از فایل input.txt.
- کلاس TestCases برای ذخیره و مدیریت مجموعه تست‌های مختلف.

تبدیل گرامرهای منظم به NFA :

در این مرحله، باید گرامرهای منظم را به یک NFA معادل تبدیل می‌کردم. بنابراین کلاس NFA را ایجاد کردم تا حالت‌ها، انتقال‌ها و وضعیت پذیرش را نگه‌داری کند. دوتا نکته زیر رو هم در نظر گرفتم:

- بررسی کردم که آیا ε در گرامر وجود دارد یا نه، تا در صورت وجود، انتقال مستقیم به پذیرش انجام شود.
- یک حالت پذیرش (F) اضافه کردم تا مقادیر پایانی پردازش شوند.

تبدیل NFA به DFA :

من در این بخش برای مدیریت مجموعه حالات و تبدیل NFA غیرقطعی به DFA قطعی، از روش پردازش مجموعه حالات (subset construction) استفاده کردم تا هر مجموعه‌ای از حالات NFA تبدیل به یک حالت واحد در DFA شود. همچنین نکات زیر رو هم درنظر گرفتم:

- به جای استفاده از یک لیست ساده برای ذخیره حالات جدید، از مجموعه (frozenset) استفاده کردم تا مانع ایجاد حالات تکراری در DFA شوم.
- برای مدیریت صف پردازش، از deque بهره بردم که عملکرد بهتری نسبت به لیست دارد.

- در تعیین حالات پذیرش، بررسی کردم که هر مجموعه‌ای که شامل یک حالت پذیرش NFA باشد، در DFA پذیرش شود.
- حالت تله رو هم درنظر گرفتم (N)

اعمال عملیات‌های مجموعه‌ای روی DFA :

در این بخش، سه عملیات اصلی را پیاده‌سازی کردم:

1. متمم → (Complement) تبدیل حالات پذیرش به حالت‌های غیرپذیرش و بالعکس.
2. اجتماع → (Union) ترکیب دو DFA به شکلی که هر دو کلمه پذیرفته شوند.
3. اشتراک → (Intersection) ترکیب دو DFA به شکلی که فقط کلماتی پذیرفته شوند که در هر دو باشند.

پیاده‌سازی main.py و اتصال بخش‌های مختلف:

در نهایت، همه این قطعات را در main.py به هم متصل کردم و نتیجه نهایی رو در فایل output.txt ذخیره کردم.