

گزارش پروژه: محاسبه ماشین تورینگ

رضا فرح بخش 9931074

معرفی

هدف از این پروژه پیاده سازی یک ماشین تورینگ است که محاسبه فرم $(3 * 5 + n!)$ را انجام می دهد، که در آن n یک عدد صحیح مثبت ارائه شده توسط کاربر است. این پروژه شامل طراحی و اجرای ماشین تورینگ و همچنین مدیریت ورودی کاربر و نمایش نتیجه محاسبه شده است. این گزارش یک نمای کلی از پروژه شامل معماری، فرمت های ورودی و خروجی و توضیح گام به گام نحوه کار ماشین تورینگ را ارائه می دهد.

معماری

این پروژه از اجزای زیر تشکیل شده است:

کلاس **TuringMachine**: این کلاس نشان دهنده ماشین تورینگ است و شامل ویژگی ها و روش های لازم برای انجام انتقال بر روی یک نوار معین است. دارای ویژگی های زیر است:

نوار: فهرستی که نوار ماشین تورینگ را نشان می دهد.

head_position: یک عدد صحیح که موقعیت فعلی سر نوار را نشان می دهد.

state: رشته ای که وضعیت فعلی ماشین تورینگ را نشان می دهد.

انتقال: فرهنگ لغت که قوانین انتقال ماشین تورینگ را تعریف می کند.

این کلاس همچنین شامل متدهای زیر است:

transition(): انتقال حالت را تا رسیدن به حالت نهایی **qf** انجام می دهد.

move_head (direction): سر نوار را در جهت مشخص شده حرکت می دهد.

تابع **account_factorial**: این تابع فاکتوریل یک عدد معین **n** را محاسبه می کند. برای محاسبه مقدار **n** استفاده می شود! در فرمول $(n!) + 5 * 3$.

بلوک کد اصلی: این بخش ورودی کاربر را کنترل می کند، نتیجه را با استفاده از تابع فاکتوریل محاسبه می کند، نوار را مقداردهی اولیه می کند، نمونه ای از کلاس **TuringMachine** ایجاد می کند و انتقال را انجام می دهد. همچنین نتیجه محاسبه شده و نتیجه به دست آمده از ماشین تورینگ را نمایش می دهد.

ورودی و خروجی

پروژه یک عدد صحیح مثبت n را به عنوان ورودی از کاربر می گیرد. از کاربر خواسته می شود که یک عدد صحیح مثبت وارد کند و برنامه از معتبر بودن ورودی اطمینان می دهد. اگر ورودی صفر یا منفی باشد، برنامه خارج می شود. فرآیند ورودی تا زمانی که یک عدد صحیح مثبت معتبر ارائه شود تکرار می شود.

خروجی پروژه شامل دو نتیجه است:

نتیجه محاسبه شده: این مقداری است که با ارزیابی عبارت $(3 * 5) + n!$ با استفاده از تابع `calcul_factorial` بدست می آید. به عنوان "[The result is: [result]" به کاربر نمایش داده می شود.

نتیجه ماشین تورینگ: این نتیجه ای است که با اجرای ماشین تورینگ روی نواری که با نتیجه محاسبه شده مقدار دهی اولیه شده است، به دست می آید. فرآیند انتقال ماشین تورینگ با استفاده از روش انتقال کلاس `TuringMachine` انجام می شود. پیکربندی نواری نهایی به صورت `The Turing machine result is: [tape"` به کاربر نمایش داده می شود.

جریان کار

```
C:\Users\Reza\PycharmProjects\FL\venv\Scripts\python.exe C:/Users/Reza/PycharmProjects/FL/turingMachine.py
Enter a positive integer (0 to exit): 1
The result is: 8
The Turing machine result is: #3
Enter a positive integer (0 to exit): 2
The result is: 11
The Turing machine result is: #3
Enter a positive integer (0 to exit): 3
The result is: 23
The Turing machine result is: #3
Enter a positive integer (0 to exit): 4
The result is: 77
The Turing machine result is: #3
Enter a positive integer (0 to exit): 5
The result is: 365
The Turing machine result is: #3
Enter a positive integer (0 to exit): 6
The result is: 2165
The Turing machine result is: #3
Enter a positive integer (0 to exit): 7
The result is: 15125
The Turing machine result is: #3
Enter a positive integer (0 to exit): 8
The result is: 120965
The Turing machine result is: #3
Enter a positive integer (0 to exit): 9
The result is: 1088645
```

این پروژه گردش کار زیر را دنبال می کند:

از کاربر خواسته می شود که یک عدد صحیح مثبت (0 برای خروج) وارد کند.

ورودی کاربر تأیید شده است، و اطمینان حاصل شود که یک عدد صحیح مثبت

است.

اگر ورودی صفر یا منفی باشد، برنامه خارج می شود.

فاکتوریل عدد ورودی با استفاده از تابع `account_factorial` محاسبه می شود.

نتیجه با استفاده از فرمول $(n! + 5 * 3)$ محاسبه می شود.

یک نوار با نتیجه محاسبه شده، با نمادهای خالی اضافی در ابتدا و انتهای مقداردهی

اولیه می شود.

نمونه ای از کلاس TuringMachine ایجاد می شود که نوار را به عنوان آرگومان ارسال می کند.

روش انتقال نمونه TuringMachine برای انجام انتقال حالت فراخوانی می شود. پیکربندی نوار نهایی به دست آمده و به عنوان نتیجه ماشین تورینگ نمایش داده می شود.

مراحل 1-9 تکرار می شود تا زمانی که کاربر یک عدد صفر یا منفی وارد کند و در این مرحله برنامه خارج می شود.

نتیجه

این پروژه با موفقیت یک ماشین تورینگ را اجرا می کند که محاسبه $(3 * 5 + n!)$ را برای یک عدد صحیح مثبت داده شده n انجام می دهد. ماشین تورینگ با استفاده از کلاس TuringMachine طراحی شده است که انتقال حالت و دستکاری نوار را مدیریت می کند. این پروژه به طور موثر ورودی کاربر را مدیریت می کند، نتیجه را محاسبه می کند و هم نتیجه محاسبه شده و هم نتیجه به دست آمده از ماشین تورینگ را در اختیار کاربر قرار می دهد. این پروژه اصول کار یک ماشین تورینگ را نشان می دهد و قدرت اتومات ها را در حل مسائل محاسباتی برجسته می کند.