ثمین مهدی پور ۹۸۳۹۰۳۹

گزارش پروژه دوم – نظریه زبان ها و ماشین ها

۱. ماشین تورینگی برای محاسبه توابع ذیل طراحی و پیاده سازی نمایید:

نکته ۱: بر مبنای رقم آخر شماره دانشجویی خود، تابعی که قرار است آن را محاسبه نمایید، انتخاب کنید.

نکته ۲: ورودی توابع ذیل را اعداد صحیح و مثبت فرض نمایید.

با توجه به شماره دانشجویی ۹۸۳۹۰۳۹:

viii. دانشجویانی که رقم آخر شماره دانشجویی‌شان '9' می‌باشد: $f(n) = (3n + 1)!$

برای پیاده سازی این ماشین از سه ماشین متوالی استفاده خواهیم کرد،

ماشین اول ۳n را محاسبه میکند،

ماشین دوم ۳n+۱ را محاسبه میکند،

و ماشین آخر فاکتوریل خروجی ماشین قبلی را محاسبه خواهد کرد.

طریق ورودی گرفتن از کاربر بصورت زیر است:

```
if __name__ == '__main__':
    flag = True
    while flag:
        n = input('Enter number : ')
        run_machine(int(n))
        c = input('If you want to continue enter 1 else 0: ')
        cont = int(c)
        if cont == 0:
            flag=False
```

هرماشین از چند بخش اصلی تشکیل شده است، بخش اولیه برای تعیین استیت ها و نقطه شروع، بخش قدم که روی نوار حرکت میکند و بخش اجرا که حرکت روی نوار را کنترل کرده و در نهایت خروجی را برمیگرداند.

همانطور که مشاهده میشود تا زمانی که کاربر بخواهد میتواند عدد وارد کند، حالا برای مثال عدد ۱ را وارد میکنیم:

```
Enter number : 1
FIRST MACHINE STARTED!
Input: 1
Tape: ['1', '0']
State: find, Sym: 1
changed tape : ['111', '0', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
State: find, Sym: 0
changed tape : ['111', '0', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
State: halt, Sym: 111
Result 3 * n: 111
```

```python
BLANK = "_"  # Symbol to represent an empty cell on the tape
tapes = {}

class TuringMachine_1:
    def __init__(self, input_number):
        print('FIRST MACHINE STARTED!')
        # Convert the input number to unary representation
        unary_representation = "1" * input_number
        print(f'Input: {unary_representation}')

        # Initialize the tape with the unary representation
        self.tape = list(unary_representation + "0")
        print(f'Tape: {self.tape}')
        self.tape.extend([BLANK] * 15)  # Extend the tape with blank symbols for working space

        # Define the states
        self.states = {
            "find",  # Find the number and repeat each '1' three times
            "halt"  # Termination state
        }
```

ماشین اول به صورت بالا تعریف میشود، لازم به ذکر است برای اینکه خروجی هرماشین به ماشین بعدی داده شود نیز نوار های خروجی را در لیست tapes ذخیره میکنیم.

```
# Define the transition function as a dictionary
self.transitions = {
    ("find", "1"): ("find", "111", "R"),  # Repeat each '1' three times
    ("find", "0"): ("halt", "0", "L")  # End of number, halt
}

self.current_state = "find"  # Start with repeating the number
self.current_position = 0  # Start at the beginning of the tape
```

استیت های اجرای ماشین اول که عملکرد ضرب در سه را انجام میدهد به صورت بالاست.

خروجی ماشین اول برای ماشین دوم فرستاده شده و خروجی ماشین دوم به صورت زیر است:

```
SECOND MACHINE STARTED!
Tape: ['1', '1', '1', '0']
State: find, Sym: 1
changed tape : ['1', '1', '1', '0', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
State: find, Sym: 1
changed tape : ['1', '1', '1', '0', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
State: find, Sym: 1
changed tape : ['1', '1', '1', '0', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
State: find, Sym: 0
changed tape : ['1', '1', '1', '1', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
State: halt, Sym: 1
Result 3 * n + 1 : 1111
```

همانطور که مشاهده میشود تنها ۱ واحد به خروجی ماشین قبلی اضافه شده، لازم به ذکر است در این ماشین تورینگ از نمایش unary استفاده کرده ایم.

استیت های ماشین دوم به صورت زیر است:

```
# Define the states
self.states = {
    "find",  # Find the number and repeat each '1' three times
    "halt"  # Termination state
}

# Define the transition function as a dictionary
self.transitions = {
    ("find", "1"): ("find", "1", "R"),  # Repeat each '1' three times
    ("find", "0"): ("halt", "1", "L")  # End of number, halt
}
```

خروجی ماشین دوم به ماشین سوم داده میشود و فاکتوریل توسط ماشین سوم محاسبه خواهد شد و به صورت نمایش unary نشان داده خواهد شد، در این ماشین ما به این صورت عمل میکنیم:

ابتدا ورودی به صورت زیر درمی آید:

```
********************************************************************************
THIRD MACHINE STARTED!
Tape: ['_', '1', '1', '1', '1', '&', '1', '_']
```

حالا با شروع کردن روی این نوار به تعداد یک های قبل از & در استک پوش میکنیم و یک های بعد از & را به اندازه استک کپی میکنیم:

```
changed tape : ['_', '1', '1', '1', '1', '&', '1', '1', '1', '1', '_']
```

با ادامه این روند در نهایت میتوانیم فاکتوریل را با نمایش UNARY محاسبه کنیم، لازم به ذکر است که تمامی استیت ها و چگونگی خواندن از نوار در کد امده است.

استیت های این ماشین به صورت زیر است:

```
# Define the transition function as a dictionary
self.transitions = {
    ('find', BLANK): ('find', BLANK, 'R'),
    ('find', '1'): ('find', '1', 'R'),
    ('find', '&'): ('change', '&', 'R'),
    ('change', '1'): ('change', '1', 'R'),
    ('change', BLANK): ('reverse', BLANK, 'L'),
    ("reverse", '1'): ("reverse", '1', 'L'),
    ("reverse", '&'): ("remove", '&', 'L'),
    ("remove", '1'): ("remove", '1', 'L'),
    ("remove", BLANK): ("star", BLANK, 'R'),
    ('star', '1'): ('find', BLANK, 'R'),
    ('halt', '#'): ('halt', BLANK, 'L')
}
```

خروجی ماشین سوم که خروجی نهایی ماشین تورینگ است بصورت زیر است:

```
changed tape : ['_', '_', '_', '_', '_', '&', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '_']
State: halt, Sym: &
Result final : 111111111111111111111111
Cause its hard to count 1s! decimal is 24
```

همانطور که مشاهده میشود ۲۴ که همان ۴! است به شیوه UNARY نمایش داده شده است.

ماشین تا زمانی که کاربر درخواست دهد قادر به محاسبه خواهد بود اما اعداد بعدی خروجی با نمایش UNARY بسیار بزرگی خواهند داشت که برای نمایش آنها زمان محاسبه طولانی خواهد بود اما ماشین قادر به انجام این محاسبات خواهد بود.

برای اطمینان n=۲ را هم تست میکنیم، خروجی نهایی بصورت زیر خواهد بود:

```
Result final : 11111111111111111111111111111111111111111111111111111111111111111111111111111111
Cause its hard to count 1s! decimal is 5040
********************************************************************************
If you want to continue enter 1 else 0:
```

نمایش با عدد یک ۷ فاکتوریل که معادل ۵۰۴۰ است بسیار طولانیست پس با نمایش طول این رشته مشاهده میکنیم که ماشین به پاسخ صحیح رسیده است.