

پروژه شماره ۳

حل مساله فاکتوریل به کمک برنامه نویسی پرولوگ

استاد: جناب آقای دکتر فیضی درخشی

دانشجو: پارسا یوسفی نژاد محمدی

شماره دانشجویی: ۱۴۰۰۵۳۶۱۱۰۴۸

هدف پروژه

در این مینی پروژه قصد داریم تا برنامه‌ای به شیوه بازگشتی بنویسیم که بررسی کند که آیا فاکتوریل عددی چون N ، برابر با F می‌شود؟ و یا از برنامه سوال کنیم تا بگوید فاکتوریل عددی مثل N ، برابر با چه خواهد بود؟ و یا اینکه عدد F فاکتوریل کدام عدد است؟. پرسیدن این دسته از کوثری‌ها هدف این پروژه بوده و در این پروژه به بررسی کد، نحوه اجرا و پاسخ کوثری‌های داده شده، خواهیم پرداخت.

تشریح پروژه

برای حل این سوال قانونی (Rule) به نام factorial تعریف میکنیم، از خاصیت بازگشتی ریاضی فاکتوریل می‌دانیم که:

$$\text{factorial}(N) = N \times \text{factorial}(N-1)$$

$$\text{factorial}(1) = 1$$

این بدین معناست که جهت محاسبه فاکتوریل N نیازمند این هستیم که فاکتوریل $N-1$ را محاسبه کنیم و همچنین برای محاسبه فاکتوریل $N-1$ باید فاکتوریل $(N-1)-1$ را محاسبه کنیم، در نهایت تنها دانسته و شرط توقف این رابطه بازگشتی، فاکتوریل 1 که همان تنها دانش ما از رابطه فاکتوریل است، می‌باشد و با جایگذاری های پسرو و نتیجه گرفتن مقادیر فاکتوریل‌ها از مقادیر قبلی‌اشان، قادر به محاسبه کردن فاکتوریل نهایی برای N خواهیم بود.

لذا تنها دانسته ما نسبت به مساله فاکتوریل به صورت مستقیم، واقعیت (fact)، **factorial(1,1)** خواهد بود، یعنی می‌دانیم که فاکتوریل عدد 1 برابر با عدد 1 خواهد بود، البته میتوانستیم شرط توقف را صفر فاکتوریل هم در نظر بگیریم، ولی تغییری در مساله ایجاد

نمی‌کرد. حال باید قانونی بنویسیم که فاکتوریل را برای مقادیر گوناگون N و F محاسبه بکند، بدین منظور از خاصیت فراخوانی‌های (call) بازگشتی در پرولوگ استفاده میکنیم، پس باید چندین شرط را بررسی کنیم که آیا فاکتوریل عددی چون N ، برابر با عددی چون F خواهد شد یا نه؟

❖ شرایط برقراری قانون $\text{factorial}(N, F)$:

- N باید عددی بزرگتر از ۱ باشد:
 - و فاکتوریل $N-1$ (M) به ازای عددی چون W درست باشد:
 - و عدد W انتخاب شده حتمن باید در این معادله صدق بکند، به طوری که:
- $N > 1,$
 $M \text{ is } N-1,$
 $\text{factorial}(M, W),$
 $F \text{ is } N \times W.$

پس همانطور که می‌بینیم برای محاسبه کردن $\text{factorial}(N, F)$ به استفاده از ویژگی بازگشتی پرلوگ متوسل شدیم؛ به پرولوگ می‌گوییم که عددی چون W را پیدا کند که حاصل ضرب آن با N برابر با F باشد، در صورتی که پرولوگ به این کوئری پاسخ true داد، متوجه میشویم که F فاکتوریل عدد N بوده است، یعنی:

$$N! = F$$

$$N! = W * (N-1)! = F$$

در روند اجرای این قانون، هنگام کوئری گرفتن از آن در ترمینال، حتمن در صورت برقرار بودن شرط اول $N > 1$ به شرط توقف کننده روند فراخوانی‌های بازگشتی خواهیم رسید، و W مقدار ۱ را خواهد گرفت، و هنگام بازگشت از فراخوانی‌های تودرتو مقدار W دائمی آپدیت می‌شود و در انتهای هر بازگشت از قانون factorial چک می‌شود که آیا F برابر با $N \times W$ است یا خیر، در صورتی که تمام این شروط نوشته شده در نهایت همگی true شوند، برنامه به ما پاسخ true را مبنی بر اینکه $N!$ برابر با F است را برمی‌گرداند.

❖ شیوه‌های کوئری دادن به برنامه:

• بررسی صحت یک سوال.

در ترمینال پرولوگ می‌توانیم بپرسیم که آیا رابطه factorial برای دو عدد دلخواه N و F برقرار است، به طوری که $N! = F$ ؟ در صورتی که چنین رابطه‌ای برقرار باشد، کامپایلر swi-prolog true را بر میگرداند، در غیر این صورت false چاپ میکند:

```
?- factorial(3,6).  
true .
```

```
?- factorial(4,25).  
false.
```

```
?- factorial(6,720).  
true .
```

```
?- factorial(120,5).  
false.
```

```
?- factorial(1,1).  
true .
```

```
?- factorial(-4,24).  
false.
```

• پرسیدن فاکتوریل یک عدد.

می‌توانیم از پرولوگ بپرسیم که فاکتوریل عدد دلخواه و معتبری چون N، برابر با چه خواهد شد. بدین منظور تنها تغییری که در هنگام کوئری دادن به پرولوگ باید انجام بدهیم، این است که بجای پارامتر دوم F متغیری چون X را جایگذاری کنیم، و از پرولوگ بخواهیم که مقدار X ای را که موجب true شدن پرسش می‌شود را بازگرداند. پرولوگ سعی میکند، در صورت امکان چنین X ای را برگرداند، اما اگر پیدا کردن $N!$ ممکن نبود، false را return خواهد کرد:

```
?- factorial(5,X).  
X = 120 .
```

```
?- factorial(8,X).  
X = 40320 .
```

```
?- factorial(1,X).  
X = 1 .
```

```
?- factorial(-3,X).  
false.
```

```
?- factorial(-12,X).  
false.
```

```
?- factorial(4,X).  
X = 24 .
```

کد پروژه

```
1 factorial(0,1).
2 factorial(1,1).
3
4 factorial(N,F):-
5     N>1,
6     M is N-1,
7     factorial(M,W),
8     F is N*W.
```

پارسا یوسفی نژاد

جمع‌بندی پروژه

در ابتدای پروژه، به تشریح رابطه ریاضی نماد و تابع factorial پرداختیم، سپس آن را به شیوه بازگشتی در زبان پرولوگ مدل کردیم و قانون و شرط توقفی (واقعیت) طراحی کردیم که بررسی کند که آیا $N!$ معادل F خواهد بود یا نه، بعد از آن به بررسی شیوه‌های پرسش سوال از برنامه با کامپایلر swi-prolog پرداختیم و هر کدام را به تفصیل بررسی کردیم. در این پروژه توانستیم با نحوه برنامه نویسی در پرولوگ آشنا شویم و برنامه‌ای بنویسیم که از خاصیت بازگشتی فراخوانی استفاده میکند.