



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

## ساختمان داده‌ها و الگوریتم‌ها

### جلسه ۳۳ : درخت دودویی عبارت

نگارنده: امین رواقی

۲۵ آذر ۱۴۰۰

## فهرست مطالب

۱	درخت دودویی عبارت
۱.۱	عبارت ریاضی میانوندی
۲.۱	تعریف بازگشتی عبارت پسوندی
۳.۱	تعریف بازگشتی عبارت پیشوندی
۴.۱	درخت عبارت

## ۱ درخت دودویی عبارت

درخت دودویی عبارت (Binary Expression Tree)، یک درخت دودویی است که برای نمایش عبارات ریاضی از آنها استفاده میشود.

درخت دودویی: بر نمایش عبارات ریاضی حاوی عملگرهای دو عملوندی و تک عملوندی.  
عبارات ریاضی: دو نوع پر استفاده این عبارت ریاضی عبارتند از:

- عبارات جبری،

- عبارات بولی.

عبارات ریاضی در حالت کلی میتوانند در چهار قالب زیر بیان شوند:

۱. عبارات میانوندی: به طور معمول ما با این نوع عبارات ریاضی سروکار داریم

۲. عبارات پسوندی: عملگر بعد از عملوند(ها) می آید،

۳. عبارات پیشوندی: عملگر قبل از عملوند(ها) می آید.

۴. درخت عبارت: که عملوندها نودهای خارجی این درخت و عملگرها نودهای داخلی این درخت را تشکیل می دهند.

در ادامه ، هر یک از نوع های مذکور را شرح می دهیم.

## ۱.۱ عبارت ریاضی میانوندی

عبارت زیر را در نظر بگیرید:

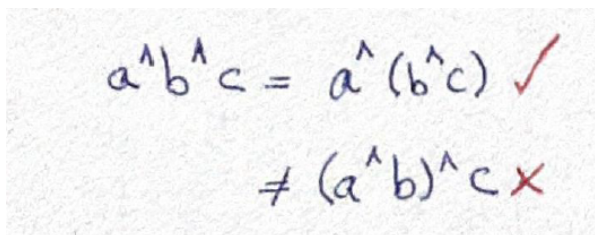
$$A^* = a + (b * c) / d - e.$$

برای محاسبه بدون ابهام چنین عبارتی میانوندی دو رویکرد زیر وجود دارد :

۱. پرائتز گذاری کامل عبارت: در این رویکرد، با توجه به معنا ( Semantic ) مورد انتظار از عبارت، آن را پرائتز گذاری کامل میکنیم،  
مثال :

$$A^* = (((a + (b * c)) / d) - e)$$

۲. ایجاد یک قرار داد برای اولیت گذاری عملگر ها: به عنوان مثال، می توان اولویت گذاری عملگرها را به ترتیب (از زیاد به کم) به صورت زیر تعیین کرد : ۱- توان از راست به چپ، ۲- ضرب و تقسیم از چپ به راست، ۳- جمع و تفریق از چپ به راست.  
مثال ۱:



مثال ۲: برای عبارت

$$A^* = a + (b * c) / d - e$$

طبق اولویت عملگرها عبارت میانوندی کامل زیر را داریم:

$$A^* = ((a + ((b * c) / d)) - e)$$

نکته تکمیلی: به طور معمول در زبان های برنامه نویسی، وجود ابهام در محاسبه یک عبارت، احتمال وجود خطای منطقی در برنامه را بالا میبرد.

یاد آوری: انواع خطاها در برنامه: ۱- خطای نحوی، ۲- خطای منطقی، ۳- خطای زمان اجرا.

تعریف بازگشتی عبارت میانوندی کامل: یک عبارت ریاضی میانوندی کامل را می توان با استفاده از یک گرامر مستقل از متن (Context Free Grammar) به صورت زیر تعریف کرد:

1  $E \rightarrow \text{Operand}$

2  $E \rightarrow ( E \alpha E )$

3  $\alpha \rightarrow /|*|+|-|...$

4  $E \rightarrow (\beta E)$

5  $\beta \rightarrow \text{Sin}|\text{cos}|...$

مثال: مراحل ساخت عبارت میانوندی کامل زیر را به کمک روابط بازگشتی فوق مشخص کنید.

$$A^* = ((a + ((b * c)/d)) - e)$$

$$E \rightarrow (EE) \rightarrow (E - E) \rightarrow (E - e) \rightarrow ((EE) - e) \rightarrow ((aE) - e) \\ \rightarrow ((aE) - e) \rightarrow \dots((a + (b * c)/d) - e)$$

نکته تکمیلی: تعریف بازگشتی فوق (گرامر مستقل از متن بالا) ضمن رعایت پراگماتیک گذاری کامل که سبب عدم ابهام در روال محاسبه می شود، این امکان را نیز فراهم میکند که اگر عبارت ریاضی نا معتبری داده شده باشد، آن را یافته و از خطای نحوی جلوگیری کند (به عبارت دیگر همزمان خطای نحوی و خطای منطقی را پوشش می دهد).

## ۲.۱ تعریف بازگشتی عبارت پسوندی

یک عبارت ریاضی پسوندی را میتوان با استفاده از یک گرامر مستقل از متن به صورت زیر تعریف کرد:

1.  $E \rightarrow \text{Operand}$
2.  $E \rightarrow EE\alpha$
3.  $\alpha \rightarrow /|*|+|-|...$
4.  $E \rightarrow E\beta$
5.  $\beta \rightarrow \text{Sin}|\text{Cos}|...$

مثال: عبارت میانوندی کامل زیر را به صورت عبارت پسوندی بنویسید.

$$A^* = ((a + ((b * c)/d)) - e)$$

$$bc*$$

$$bc * d/d$$

$$abc * d/+$$

در نهایت عبارت پسوندی زیر حاصل می شود:

$$abc * d/+ e -$$

عبارت پسوندی فوق را میتوان برای تشخیص معتبر بودن به کمک گرامر فوق پویش کنیم:

$$E \rightarrow EE \rightarrow (EE-) \rightarrow (Ee-) \rightarrow (EEe-) \rightarrow \dots$$

## ۳.۱ تعریف بازگشتی عبارت پیشوندی

یک عبارت ریاضی پیشوندی را میتوان با استفاده از یک گرامر مستقل از متن به صورت زیر تعریف کرد:

1.  $E \rightarrow \text{Oprand}$
2.  $E \rightarrow \alpha EE$
3.  $\alpha \rightarrow /|*|+|-|...$
4.  $E \rightarrow \beta E$
5.  $\beta \rightarrow \text{Sin}|\text{Cos}|...$

مثال : عبارت میانوندی کامل زیر را به صورت عبارت پیشوندی بنویسید.

$$A * = ((a + ((b * c) / d)) - e)$$

$*bc$

$*bcd$

$+a / *bcd$

در نهایت عبارت پیشوندی زیر حاصل می‌شود:

$- + a / *bcde$

عبارت پیشوندس فوق را میتوان برای تشخیص معتبر بودن به کمک گرامر فوق پویش کنیم:

$$E \rightarrow E * E \mid E / E \mid E + E \mid E - E \mid (E) \mid \dots \rightarrow - + a / *bcde$$

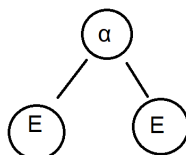
**نکته مهم:** در عبارت پسوندی و پیشوندی پرانتز گذاری نداریم و در عین حال عبارت بدین ابهامی هم برای محاسبه در اختیار داریم . با این وجود ، برای عبارت میانوندی دیدیم که اگر عبارت با پرانتز گذاری کامل نداشته باشیم ، پتانسیل ابهام در آن وجود دارد.

## ۴.۱ درخت عبارت

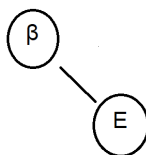
هر عبارت میانوندی کامل را میتوان با یک درخت دودویی نمایش داد. از آنجاییکه حالت مصور، کمک شایانی در فهم مطالب می‌کند، پرداختن به آن حایز اهمیت است.

مراحل ساخت بازگشتی یک درخت دودویی از عبارت میانوندی کامل به صورت زیر است :

۱. اگر  $E\alpha E$  باشد داریم:



۲. اگر  $\beta E$  باشد ، آنگاه داریم:

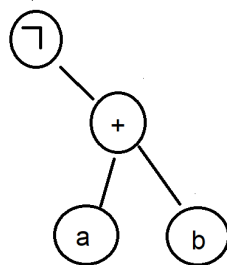


**نکته:** به صورت پیشفرض E را در سمت راست عملگر تک عملوندی قرار می‌دهیم.

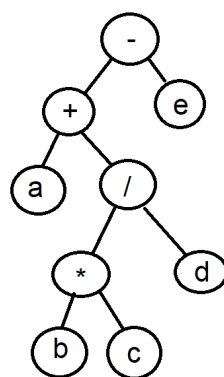
۳. اگر Operand باشد ، آنگاه داریم:



مثال ۱: درخت دودویی عبارت مربوط به عبارت ریاضی  $A^* = \neg(a + b)$  را ترسیم کنید.



مثال ۲: درخت دودویی عبارت مربوط به عبارت ریاضی  $A^* = ((a + ((b * c)/d)) - e)$  را ترسیم کنید.



**نکته مهم:** پیمایش پیشوندی، پسوندی و میانوندی درخت دودویی عبارت به ترتیب معادل عبارت پیشوندی، عبارت پسوندی و عبارت میانوندی می‌باشد.

**تمرین:** تبدیل عبارت میانوندی کامل، پیشوندی، پسوندی و درخت دودویی عبارت به یکدیگر به چه نحو قابل انجام است؟