

## تمرین چهارم

### اصول طراحی کامپایلرها

#### تمرین ۱

برای زبان های زیر عبارات منظم بنویسید.  
(آ) اعداد باینری که از ۱۰۱ بزرگ تر باشند.  
(ب) اعداد دهدهی که بین ۰ تا ۲۵۵ باشند. توجه کنید که صفر های قبل از عدد پذیرفته نیستند.

#### تمرین ۲

گرامر زیر را در نظر بگیرید.

$$E \rightarrow E * E \mid E + E \mid (E) \mid \text{int}$$

(آ) چند درخت تجزیه ی متمایز می توان برای عبارت  $5 * 3 + (2 * 7) + 4$  رسم کرد؟ درخت ها را رسم نمایید.  
(ب) وجود کدام یک از قوانین زیر برای یکتا بودن درخت تجزیه، لازم است؟ (ممکن است چند قانون لازم باشد)

(الف) اولویت پرانتز بیش تر از ضرب باشد.	(ب) جمع از چپ شرکت پذیر باشد.
(ج) ضرب از چپ شرکت پذیر باشد.	(د) اولویت ضرب از جمع بیش تر باشد.

#### تمرین ۳

گرامر زیر را برای زیرمجموعه ای از جملات انگلیسی در نظر بگیرید که در آن NP، VP، AP و S نمادهای ناپایانه و noun، adjective و verb نمادهای پایانه هستند. S نماد ابتدایی است.

$$S \rightarrow NP VP \mid NP VP NP$$

$$NP \rightarrow \text{the AP noun}$$

$$AP \rightarrow AP \text{ adjective} \mid \epsilon$$

$$VP \rightarrow \text{verb}$$

(آ) یک گرامر  $LL(1)$  ارائه کنید که همین زبان را تولید کند.

(ب) جدول تجزیه گرامر خود را بسازید.

#### تمرین ۴

برای قاعده ی داده شده به روش Back patch کد میانی به زبان C بنویسید. فرض کنید آدرس اولین دستورالعملی که تولید می شود ۱۰۰ باشد.

$$(a < b \text{ and } c < d) \text{ or } e < f$$

## تمرین ۵

برنامه‌ی زبان پاسکال که در ذیل آمده است را به فرم‌های میانی خواسته شده تبدیل نمایید.

```
while (A < E) do begin
    while (E - A < G) do E := A + F;
    G := A + F
end
```

آ) علامت پسوندی

ب) چهارتایی

ج) سه تایی

د) سه تایی غیر مستقیم

## تمرین ۶

گرامر زیر نوع خاصی از پرانتز را توصیف می کند.

```
A → ε | A + B | C + C
B → C | C * B
C → (A) | int
```

A, B و C ناپایانه هستند و  $+$ ,  $*$ ,  $($ ,  $)$  و `int` پایانه هستند.

آ) این گرامر را از سمت چپ فاکتور بگیرید.

ب) مجموعه‌ی ابتدا و پیرو را برای هر ناپایانه موجود در گرامر به دست آمده در قسمت آ بنویسید.

ج) آیا در گرامر اصلی مغایرتی وجود دارد؟ اگر جواب مثبت است، مشخص کنید که این مغایرت از چه نوعی است و بین کدام دو نماد به وجود می آید؟

د) با استفاده از این معلومات یک جدول تجزیه‌ی LL برای گرامر به دست آمده در قسمت آ بسازید.

ه) فرض کنید با استفاده از جدول قسمت د، یک تجزیه‌گر LL برای گرامر فوق ساختیم، رشته‌ی زیر را در نظر بگیرید.

$((int + int) + int) + int +$

مشخص کنید هنگام تجزیه‌ی این رشته چه خطایی رخ می دهد؟ زمان رخ دادن خطا چه مقدار از رشته پویش شده است؟ تجزیه‌گر در این زمان چه عملکردی دارد؟

## تمرین ۷

گرامرهای  $LR(1)$ ، تمامی گرامرهای  $LL(1)$  و بسیاری دیگر از گرامرها را در بر می گیرد ولی گرامرهای  $LL(1)$  و  $LR(0)$  قابل مقایسه نیستند. به عبارت دیگر هیچ یک زیر مجموعه‌ای از دیگری نیست. به سوالات زیر در مورد گرامرهای LR و LL پاسخ دهید.

آ) آیا یک گرامر LR می‌تواند مبهم باشد؟

ب) گرامری را مثال بزنید که  $LL(1)$  باشد ولی  $LR(0)$  نباشد. توضیح دهید که چرا این گرامر این ویژگی را دارد.

ج) یک گرامر غیر مبهم ارائه کنید که  $LR(1)$  نباشد. توضیح دهید که چرا این گرامر این ویژگی را دارد.

د) یک گرامر  $LL(2)$  ارائه کنید که  $LL(1)$  نباشد. توضیح دهید که چرا این گرامر این ویژگی را دارد.

## تمرین ۸

گرامر زیر را در نظر بگیرید:

$$S \rightarrow aAB$$

$$A \rightarrow bA \mid \varepsilon$$

$$B \rightarrow cb$$

یک تجزیه‌گر نزولی بازگشتی<sup>۱</sup> برای این گرامر بنویسید. فرض کنید توابع مورد نیاز (مثلا برای خواندن ورودی) را دارید.

### توضیحات:

هرگونه شباهت بین تمرین‌های تحویلی نمره‌ی منفی خواهد داشت.

سوالاتی از تمرین ۳ که تدریس نشده بودند را می‌توانید همزمان با تمرین ۴ تحویل دهید.

مهلت تحویل این تمرین تا ۳ دیماه خواهد بود و با توجه به منعطف نبودن برنامه‌ی تمرین‌ها، این تاریخ به هیچ عنوان تمدید

نخواهد شد. لذا از ایمیل زدن به تدریس یار درس، جهت تمدید جدا خودداری نمایید.

.

موفق باشید

---

<sup>1</sup> Recursive-Descent Parser