

## تمرین سوم

### اصول طراحی کامپایلرها

#### تمرین ۱

به سوالات زیر پاسخ کوتاه دهید.

(آ) چک کردن نوع متغیرها در کدام فاز انجام می‌شود؟

(ب) پردازش ماکرو در کدام فاز انجام می‌شود؟

(ج) پردازش کاراکتر به کاراکتر کد در کدام فاز انجام می‌شود؟

(د) اگر رشته‌ی ورودی از چپ به راست پویش شود، هنگام تجزیه‌ی رشته‌ی ورودی، تجزیه‌گر بالا به پایین از چه نوع اشتقاقی استفاده می‌کند؟

(ه) در یک کامپایلر، صفحه کلید (حروف) یک زبان در چه فازی شناسایی می‌شوند؟

(و) در یک کامپایلر، ساختمان داده‌ای که متغیرها و ویژگی‌های مربوط به آن‌ها را نگهداری می‌کند، چه نام دارد؟

(ز) چرا در تحلیل لغوی از بافر استفاده می‌شود؟

(ح) منظور از هرس دسته<sup>۱</sup> در تجزیه‌ی پایین به بالا چیست؟

#### تمرین ۲

(آ) نشان دهید تمام رشته‌های دودویی که با گرامر زیر تولید می‌شوند، مقادیر قابل قسمت بر ۳ دارند یا خیر. راهنمایی: از استقرای ریاضی روی تعداد گره‌های درخت تجزیه استفاده کنید.

$\text{num} \rightarrow 11 \mid 1001 \mid \text{num } 0 \mid \text{num num}$

(ب) آیا این گرامر، تمامی رشته‌های دودویی قابل تقسیم بر ۳ را تولید می‌کند؟

#### تمرین ۳

(آ) گرامر زیر یک زبان منظم را توصیف می‌کند.

$E \rightarrow T b E \mid T$

$T \rightarrow a$

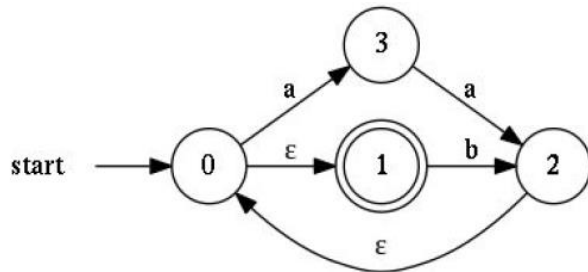
عبارت منظمی بسازید که بر این زبان منطبق باشد. از نشانه گذاری‌های اصلی زیر استفاده کنید:

$\varepsilon, a, b, AB, A|B, (A), A^*$

---

<sup>1</sup> Handle Pruning

ب) عبارت منظم  $c \mid (a^*b)$  را در نظر بگیرید. NFA ای بسازید که این زبان را بپذیرد.  
 ج) NFA زیر را در نظر بگیرید.

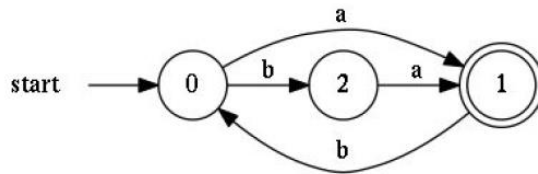


با استفاده از ساخت زیر مجموعه، DFA معادل را رسم نمایید.  
 د) گرامر قسمت آ را در نظر بگیرید. قانون تولید E را به شکل زیر تغییر می‌دهیم.

$$E \rightarrow T b E \mid T \mid a E a$$

بقیه‌ی قوانین تغییری نمی‌کند. آیا گرامر جدید منظم است؟ اگر پاسخ مثبت است، برای این زبان، یک عبارت منظم بنویسید. اگر پاسخ منفی است، دلیل را بیان کنید.

ه) DFA زیر را در نظر بگیرید. گرامر مستقل از متنی ارائه دهید که زبان مشابهی را توصیف کند.



#### تمرین ۴

پویش از راست به چپ به این ترتیب است که ابتدا، آخرین کاراکتر از رشته‌ی ورودی وارد ماشین می‌شود و سپس کاراکتر یکی مانده به آخر وارد ماشین می‌شود و این روند تا انتهای رشته ادامه می‌یابد. به عبارت دیگر از آن جا که رشته را از انتها پویش می‌کنیم، ماشین تطبیق‌ی که می‌سازیم باید یک رشته‌ی ورودی را با یک الگوی معکوس تطبیق دهد.

آ) یک الگوریتم برای تبدیل عبارات منظم به NFA هایی که به صورت معکوس یک رشته را با یک عبارت منظم تطبیق می‌دهند، معرفی کنید. اثبات کنید که این الگوریتم درست کار می‌کند.

ب) یک مثال از مجموعه‌ای از عبارات‌های منظم و یک رشته بیاورید به طوری که اگر رشته را از چپ به راست پویش کنیم، نشانه‌های متفاوتی در مقایسه با پویش از راست به چپ ایجاد شود. فرض کنید که برای اجتناب از ایجاد تضاد، هر رشته را با طولانی‌ترین عبارت منظم تطبیق می‌دهیم.

## تمرین ۵

گرامر مستقل از متن زیر را در نظر بگیرید. ( قواعد نحوی این گرامر تقریباً مشابه قواعد نحوی لیست ها در زبان برنامه نویسی LISP است ) S نماد شروع او ( , a پایانه است.

$S \rightarrow ()$   
 $S \rightarrow a$   
 $S \rightarrow (A)$   
 $A \rightarrow S$   
 $A \rightarrow A, S$

آ ( به طور دقیق نشان دهید چرا این گرامر یک گرامر LL(1) نیست؟  
ب ( گرامر را به گونه ای بازنویسی کنید که برای تجزیه ی بازگشتی نزولی مناسب باشد.  
ج ( رویه ی بازگشتی S(، مربوط به تجزیه گر بازگشتی نزولی گرامر به دست آمده در قسمت ب را بنویسید.

## تمرین ۶

با فاکتور گیری از سمت چپ و یا از بین بردن بازگشت از چپ، گرامر های زیر را به گرامر های LL(1) تبدیل کنید .  
فرض کنید گرامر ها مبهم نیستند .  
آ (

$A \rightarrow ABC \mid BC \mid C$   
 $B \rightarrow \text{int} \mid (A)$   
 $C \rightarrow \text{bool}$

ب (

$L \rightarrow \text{bool} \mid \text{bool} + \text{bool} \mid \text{bool} + \text{int}$

ج (

$A \rightarrow \text{int} \mid A + \text{int} \mid \text{int} - A \mid A - (A)$

## تمرین ۷

حلقه ی for در پاسکال استاندارد چنین است:

for v := initial to final do stmt

که معادل است با کد زیر:

Begin

$t_1 := \text{initial} ; t_2 := \text{final};$

if  $t_1 \leq t_2$  then begin

$v := t_1 ;$

    stmt;

while  $v \neq t_2$  do begin

$v := \text{succ}(v);$

    stmt;

end

end

end

الف) برنامه‌ی پاسکال زیر را در نظر بگیرید:

Program forloop (input , output) ;

var i, initial, final : integer ;

begin

read (initial , final);

for i := initial to final do

    writeln (i)

end

چنانچه  $\text{initial} = \text{MAXINT} - 5$  و  $\text{final} = \text{MAXINT}$  باشد به طوریکه MAXINT بزرگترین عدد صحیحی است که ماشین هدف می‌پذیرد، این برنامه چگونه عمل می‌کند؟

ب) یک تعریف نحو هادی ارائه دهید که حلقه‌ی for زبان پاسکال را به کد سه آدرسه‌ی صحیح ترجمه نماید.

تمرین ۸

الف) برای گرامر زیر با استفاده از روش تجزیه‌ی بازگشتی نزولی، کد میانی به صورت چهارتایی تولید کنید.

$\text{Exp} \rightarrow \text{Exp} + \text{Term}$

$\text{Exp} \rightarrow \text{Exp} - \text{Term}$

$\text{Exp} \rightarrow \text{Term}$

$\text{Term} \rightarrow \text{Term} * \text{Factor} \mid \text{Term} / \text{Factor}$

$\text{Term} \rightarrow \text{Factor}$

Factor  $\rightarrow$  (Exp) | id

ب) برنامه‌ی خود را روی ورودی  $A / (B - C)$  تست کنید.

ج) دنباله‌ی توابعی که در برنامه صدا زده شده و از آنها برگشت انجام می‌پذیرد را مشخص کرده و برای هر رویه اطلاعات معنایی که منتقل می‌نماید و چهارتایی که تولید می‌کند را مشخص نمایید.

### توضیحات:

هرگونه شباهت بین تمرین‌های تحویلی نمره‌ی منفی خواهد داشت.

مهلت تحویل این تمرین تا ۱۹ آذر ماه 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.

در صورتی که بخشی از مباحث مورد سوال تا مهلت تحویل تدریس نشده باشند، فقط سوالات مربوطه به این مباحث را می‌توانید با تمرین ۴ تحویل دهید.

موفق باشید