دانشکده کامپیوتر

تمرین سری دوم

دکتر سعید پارسا

الناز رضایی

فروردین ۱۴۰۱

سوال ۱:

یک گرامر مستقل از متن برای اعداد رومی ارائه دهید.

پاسخ ۱:

در زیر گرامر مستقل از متن برای اعداد رومی از ۱ تا ۱۰۰۰ نوشته شده است:

$S \rightarrow ABCD$

$A \rightarrow m|mm|mmm|\lambda$

$B \rightarrow cd|cm|E|dE$

$C \rightarrow xl|xc|F|lF$

$D \rightarrow iv|ix|G|vG$

$E \rightarrow c|cc|ccc|\lambda$

$F \rightarrow x|xx|xxx|\lambda$

$G \rightarrow i|ii|iii|\lambda$

سوال ۲:

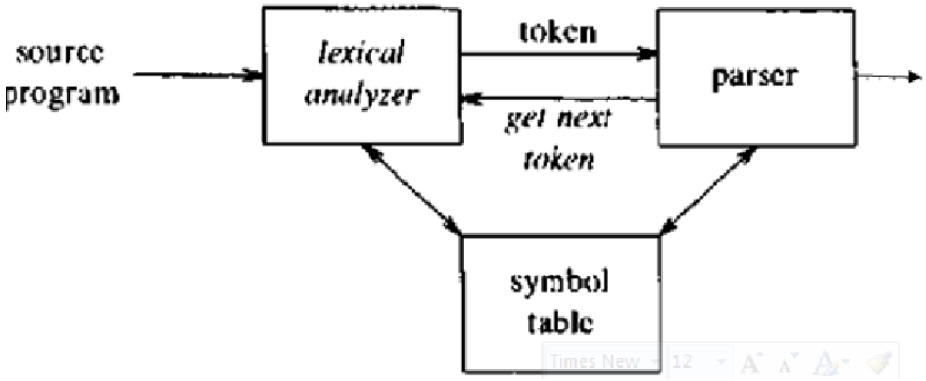
آیا امکان ترکیب analyzer lexical و parser وجود دارد؟ با ذکر دلیل توضیح دهید.

پاسخ ۲:

بله، امکان ترکیب آن‌ها وجود دارد. چرا که Lexical Analyzer کاراکتر به کاراکتر خوانده و آن‌ها را tokenize می‌کند و parser از آن برای Syntax Analysis که تنها ترتیب را چک می‌کند و معنا در آن مهم نیست، استفاده می‌کند. در واقع در parsing، با در دست داشتن یک look ahead، تشخیص می‌دهد که term بعدی، چه میانی، چه غیر میانی می‌آید. بنابراین در روش parsing، همیشه انتظار یک چیزی را داریم.

از مزایای این روش می‌توان به انجام شدن parsing و tokenization در یک گام اشاره کرد و تنها یک metalanguage مورد نیاز است و به ساختار واژگان بی‌قاعده راحت‌تر می‌توان رسیدگی کرد و گرامرهای ترکیبی را با این روش آسان‌تر می‌توان مورد بررسی قرار داد.

همچنین از معایب آن می‌توان به سخت‌تر شدن دیباگ و درک parser تولید شده اشاره کرد. به علاوه این parser ترکیبی، از نظر زمان و حافظه بهینگی کمتری دارد.



سوال ۳:

گرامر مستقل از متن زیر را در نظر بگیرید:

$$S \rightarrow SS+ | SS* | a$$

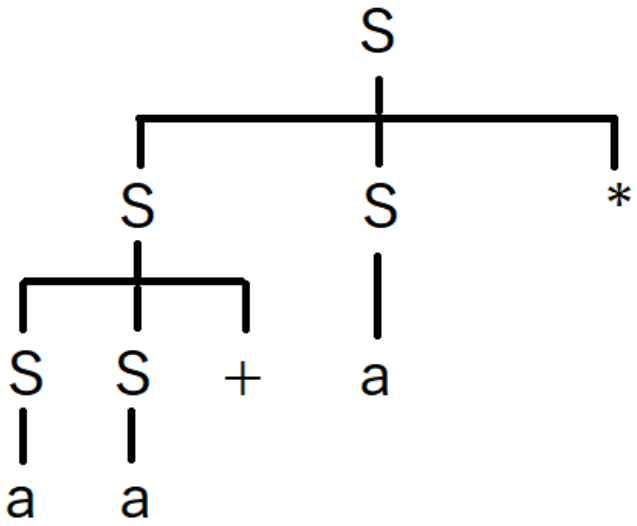
نشان دهید رشته aa+a* چگونه توسط این گرامر تولید می شود؟ tree parse مربوط به این رشته را رسم کنید.

پاسخ ۳:

Parsing stack	Source file	Action
S	aa+a*	SS*
*SS	aa+a*	SS+
*S+SS	aa+a*	a
*S+Sa	aa+a*	delete
*S+S	a+a*	a
*S+a	a+a*	delete
*S+	+a*	delete
*S	a*	a
*a	a*	delete
*	*	delete

$$S \rightarrow SS* \rightarrow SS+S* \rightarrow aS + S* \rightarrow aa + S* \rightarrow aa + a*$$

طبق جدول و عبارت بالا، نتیجه می‌گیریم که رشته ورودی داده شده، توسط این گرامر پذیرفته می‌شود.


S
S
S
*
S
S
+
a
a
a

سوال ۴:

یک تحلیلگر لغوی از الگو های زیر جهت تشخیص سه توکن T۱، T۲، T۳ روی الفبای a، b، c استفاده می کند:

T1: a?(b|c)*a

T2: b?(a|c)*b

T3: c?(b|a)*c

اگر 'x؟' به معنای اتفاق ۰ یا ۱ بار نماد x باشد و تحلیلگر لغوی طولانی ترین پیشوند ممکن را به عنوان یک توکن در خروجی تولید کند، ورودی bbaacabca به تحلیلگر لغوی چه ترتیب از توکن ها را از چپ به راست تولید خواهد کرد؟

پاسخ ۴:

T1: a(b|c)*a + (b|c)*a

T2: b(a|c)*b + (a|c)*b

T3: c(b|a)*c + (b|a)*c

bbaac (longest prefix) $\rightarrow T3$

abca (remaining part) $\rightarrow T1$

The answer is : T3T1

سوال ۵:

زبان تولید شده توسط گرامر های زیر را بیان کنید:

$S \rightarrow 0S1|01$

$S \rightarrow S(S)S|\epsilon$

پاسخ ۵.۱:

$L(G) = w \mid w$ is a string of equal numbers of zeroes and ones in order

or

$L(G) = (0)^n(1)^n, n \geq 1$

پاسخ ۵.۲:

$L(G) = w \mid w$ is a combination of nested brackets which each starting bracket, has an ending bracket.(e.g. (())())

در واقع، این گرامر ترکیبی از پرانتزها را ارائه می‌کند که در داخل یک پرانتز بزرگ، مجموعه‌ای از پرانتزها وجود دارد که هر پرانتز بازی، یک پرانتز بسته نیز دارد و داخل هر یک از این پرانتز های بزرگ، پرانتزهای داخلیش حتما باید قبل بسته شدن پرانتز بزرگ بسته شده باشند.
(در اینجا منظور از پرانتز بزرگ، این نیست که یک پرانتز از بقیه بزرگتر باشد، بلکه منظور یک پرانتز کلی است که داخلش پرانتز های دیگری وجود دارد.)