

آزمون پایان ترم درس کامپایلر

دانشگاه صنعتی خواجه نصیرالدین طوسی - دانشکده ریاضی

دی ماه ۱۴۰۲ - مدت زمان آزمون: ۹۰ دقیقه

۱- گرامر زیر را در نظر بگیرید:

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow \times FT' \mid \epsilon$$

$$F \rightarrow (E) \mid id$$

الف) برای هر یک از nonterminalها، مجموعه FIRST و FOLLOW را به دست آورید.

ب) بدون استفاده از جدول تحلیل نحوی (parsing table)، نشان دهید گرامر داده شده از نوع LL(1) است.

پ) جدول تحلیل نحوی LL(1) زیر را برای گرامر داده شده تکمیل نمایید.

NONTERMINAL	INPUT					
	<i>id</i>	+	\times	()	\$
<i>E</i>						
<i>E'</i>						
<i>T</i>						
<i>T'</i>						
<i>F</i>						

ت) روند کامل چگونگی تحلیل نحوی LL(1) را برای رشته ورودی $id * (id)$ در جدول زیر نشان دهید. توجه کنید که بالای پشته در سمت چپ قرار دارد. همچنین، در ستون Action، دقیقاً یکی از دو مورد قانون به کار رفته برای اشتقاق و یا رخداد تطابق (match) باید ذکر گردد. ستون Matched نیز زیررشته‌ای از رشته ورودی را نشان می‌دهد که تاکنون با اعمال انجام شده در ستون Action قابل اشتقاق است.

MATCHED	STACK	INPUT	ACTION
	$E\$$	$id * (id)\$$	

۲- گرامر توسعه یافته (augmented grammar) زیر را در نظر بگیرید:

شماره قانون	قانون
0	$E' \rightarrow E$
1	$E \rightarrow id[E]$
2	$E \rightarrow id$

الف) اتوماتون LR(0) items را برای گرامر توسعه یافته داده شده رسم نمایید.

ب) جدول تحلیل نحوی زیر (که متناظر با اتوماتان LR(0) items به دست آمده در بخش الف است) را تکمیل کنید. شماره گذاری حالت های اتوماتون را از عدد ۱ شروع کنید.

شماره حالت	ACTION				GOTO
	id	[]	\$	E

پ) با استفاده از جدول بخش ب، آیا گرامر داده شده از گونه LR(0) است؟

ت) نشان دهید گرامر داده شده از گونه SLR(1) است و جدول تحلیل نحوی متناظر با آن را رسم کنید.

ث) با استفاده از جدول تحلیل نحوی SLR(1)، روند کامل چگونگی تحلیل نحوی را برای رشته $id[id[id]]$ در جدول زیر نشان دهید. توجه کنید که بالای پشته حالت ها و پشته نمادها در سمت راست قرار دارند.

STACK OF STATES	STACK OF SYMBOLS	INPUT	ACTION
\$1	\$	$id[id[id]]$ \$	

۳- اگر هر یک از قانون های گرامر G به شکل $A \rightarrow BC$ یا $A \rightarrow a$ باشد که در آن A, B و C از گونه nonterminal و a از گونه terminal هستند، آنگاه گفته می شود که G به شکل نرمال چامسکی (CNF) است. هیچیک از قانون های G منجر به تولید رشته تهی (ϵ) نمی گردد، مگر حالتی که رشته تهی عضوی از زبان گرامر تولید شده باشد (یعنی $\epsilon \in \mathcal{L}(G)$). مرحله های تبدیل یک گرامر دلخواه غیر حساس به متن به گرامری معادل به شکل نرمال چامسکی را به ترتیب بیان نمایید.

۴- الف) به طور مختصر مزیت به کارگیری روش Backpatching در تولید کد میانی را بیان کنید.

ب) اکنون، فرض کنید شیوه تولید کد میانی برای عبارت‌های بولی، با روش Backpatching به شکل زیر باشد. نماد *nextinstr* بیانگر شماره دستورالعمل بعدی است که قرار است تولید شود یا بر روی آن عملی صورت پذیرد. مقدار اولیه این نماد برابر با ۱۰۰ است. همچنین، نماد *rel* بیانگر عملگرهای مقایسه‌گر همانند < و = است. توجه کنید که تحلیل نحوی و تولید کد به شیوه پایین به بالا (bottom-up) صورت می‌گیرد.

$B \rightarrow B_1 \text{ or } M B_2$	{ <i>backpatch</i> (<i>B</i> ₁ . <i>falselist</i> , <i>M.instr</i>); <i>B.true</i> <i>list</i> = <i>merge</i> (<i>B</i> ₁ . <i>true</i> <i>list</i> , <i>B</i> ₂ . <i>true</i> <i>list</i>); <i>B.false</i> <i>list</i> = <i>B</i> ₂ . <i>false</i> <i>list</i> ; }
$B \rightarrow B_1 \text{ and } M B_2$	{ <i>backpatch</i> (<i>B</i> ₁ . <i>true</i> <i>list</i> , <i>M.instr</i>); <i>B.true</i> <i>list</i> = <i>B</i> ₂ . <i>true</i> <i>list</i> ; <i>B.false</i> <i>list</i> = <i>merge</i> (<i>B</i> ₁ . <i>false</i> <i>list</i> , <i>B</i> ₂ . <i>false</i> <i>list</i>); }
$B \rightarrow \text{not } B_1$	{ <i>B.true</i> <i>list</i> = <i>B</i> ₁ . <i>false</i> <i>list</i> ; <i>B.false</i> <i>list</i> = <i>B</i> ₁ . <i>true</i> <i>list</i> ; }
$B \rightarrow (B_1)$	{ <i>B.true</i> <i>list</i> = <i>B</i> ₁ . <i>true</i> <i>list</i> ; <i>B.false</i> <i>list</i> = <i>B</i> ₁ . <i>false</i> <i>list</i> ; }
$B \rightarrow E_1 \text{ rel } E_2$	{ <i>B.true</i> <i>list</i> = <i>makelist</i> (<i>nextinstr</i>); <i>B.false</i> <i>list</i> = <i>makelist</i> (<i>nextinstr</i> + 1); <i>gen</i> ('if' <i>E</i> ₁ . <i>addr</i> <i>rel</i> <i>E</i> ₂ . <i>addr</i> 'goto -'); <i>gen</i> ('goto -'); }
$B \rightarrow \text{true}$	{ <i>B.true</i> <i>list</i> = <i>makelist</i> (<i>nextinstr</i>); <i>gen</i> ('goto -'); }
$B \rightarrow \text{false}$	{ <i>B.false</i> <i>list</i> = <i>makelist</i> (<i>nextinstr</i>); <i>gen</i> ('goto -'); }
$M \rightarrow \epsilon$	{ <i>M.instr</i> = <i>nextinstr</i> ; }

حال، با توجه به شیوه تولید کد میانی داده شده، درخت تحلیل نحوی را برای عبارت بولی زیر رسم نمایید.

$$(a < b \text{ and } c < d) \text{ or } (\text{not } (e = f))$$

پ) برای درخت تحلیل نحوی قسمت ب، مقدار صفت‌های *true**list* و *false**list* را برای نودهای از گونه *B* و همچنین مقدار صفت *instr* را برای نودهای از گونه *M* به دست آورید.

ت) کد تولید شده به دست آمده از پیمایش درخت تحلیل نحوی را بنویسید. مقصد پرش چه تعدادی از کدهای تولید شده دارنده دستور *goto* مشخص است؟

۵- مشخص کنید کدامیک از عبارتهای زیر درست و کدامیک نادرست است؟ (ذکر دلیل لازم نیست).

الف) روش تحلیل نحوی بالا به پایین (top-down parsing) را می‌توان یک چپ‌ترین اشتقاق در جهت عکس (leftmost derivation in reverse) از یک رشته ورودی با گرامر داده شده تلقی کرد.

ب) روش recursive-descent parsing حالت خاصی از روش predictive parsing است.

پ) هر گرامر SLR(1) حتماً غیرمبهم است و هر گرامر غیرمبهم حتماً SLR(1) است.

ت) فرض کنید گرامر زیر را G و گرامر به دست آمده از انجام عمل left-factoring روی G را G' بنامیم. در این صورت، تعداد قانون‌های G و G' با هم برابر خواهند بود.

$$A \rightarrow Am \mid aA \mid An \mid aaB$$

$$B \rightarrow b$$