

تکلیف دوم درس طراحی کامپایلرها - ترم ۲-۹۸۹۹

لطفاً سوالها را به صورت تاپی یا اسکن دست نوشته تمیز آماده و با نام name_family_stdno_hw2 ذخیره کرده در سامانه درس آپلود کنید.

۱- در مورد گرامر و رشته ورودی زیر به سؤالاتی که در ادامه آمده است پاسخ دهید:

$$S \rightarrow S + S \mid S S \mid (S) \mid S * \mid a \text{ with string } (a + a) * a.$$

- الف) یک اشتقاق چپ (left-most derivation) برای رشته ورودی بنویسید
ب) یک اشتقاق راست (right-most derivation) برای رشته ورودی بنویسید
ج) یک درخت تجزیه برای این رشته بنویسید
د) آیا گرامر مبهم یا غیر مبهم است؟ توضیح دهید
ه) زبانی که این گرامر تولید میکند را شرح دهید.

۲- گرامر زیر را به گرامری تبدیل کنید که بازگشتی چپ مستقیم یا ضمنی نداشته باشد

$$\begin{aligned} A &\rightarrow Ba \mid Aa \mid c \\ B &\rightarrow Bb \mid Ab \mid d \end{aligned}$$

۳- گرامر زیر تلاشی جهت نوشتن گرامر عبارات شرطی بدون ابهام است. آیا ابهام برطرف شده است؟ اگر هنوز گرامر مبهم است نشان دهید که ابهام وجود دارد.

$$\begin{aligned} stmt &\rightarrow \text{if } expr \text{ then } stmt \\ &\quad \mid matchedStmt \\ matchedStmt &\rightarrow \text{if } expr \text{ then } matchedStmt \text{ else } stmt \\ &\quad \mid other \end{aligned}$$

۴- برای گرامر زیر، یک sudocode برای اجرای الگوریتم Recursive Descent به صورت بازگشتی بنویسید.

$$\begin{aligned} S &\rightarrow S \text{ and } S \\ &\quad \mid S \text{ or } S \\ &\quad \mid T \\ &\quad \mid a \\ T &\rightarrow a \end{aligned}$$

۵- الف) بررسی کنید آیا می توان برای گرامر زیر یک پارسر LL1 طراحی کرد؟ چرا؟

$$A \rightarrow Ax \mid aA \mid Ay \mid aab$$

ب) تحقیق و بررسی کنید شروط لازم و کافی برای LLK بودن یک گرامر چیست و آنها را بیان کنید. سپس بررسی کنید آیا برای گرامر فوق پارسر LLK می توان طراحی کرد یا نه؟ اگر پاسخ شما بله است برای کدام kها؟

ج) در صورتی که پاسخ شما به قسمت اول خیر است تلاش کنید برای گرامر فوق یک گرامر معادل طراحی کنید که LL1 باشد.

د) با توجه به مثال فوق بررسی کنید آیا می توان برای هر گرامر LLk یک گرامر معادل LL1 طراحی کرد؟ علت پاسخ خود را توضیح دهید.

(سؤال اختیاری) گرامر زیر را در نظر بگیرید و فرض کنید e, s ترمینال هستند. جهت حل ابهامی که در بسط اختیاری $else$ (در غیرپایانی $stmtTail$) پیش می‌آید، هر جا از ورودی $else$ دیدیم آن را مصرف می‌کنیم. بدین ترتیب می‌توانیم یک تجزیه‌گر پیشگو (predictive parser) بسازیم. با استفاده از ایده سمبلهای هماهنگ‌ساز (synchronizing) به سؤالات زیر پاسخ دهید.

$stmt$	→	if e then $stmt\ stmtTail$	الف) برای این گرامر یک جدول تجزیه بسازید که خطاها را نیز اصلاح کند.
		while e do $stmt$	
		begin $list$ end	
		s	
$stmtTail$	→	else $stmt$	ب) عملیات تجزیه‌گر خود را برای دو نمونه ورودی دلخواه تست کنید که در یک ورودی خطایی باشد که با $follow$ سمبل روی استک کنترل شود و در دیگری با $first$ ساختار سطح بالاتر.
		ϵ	
$list$	→	$stmt\ listTail$	
$listTail$	→	$; list$	
		ϵ	