

آمادگی برای آزمون میانی طراحی کامپایلر

(۱) عبارت منظم داده شده را به NFA و سپس به DFA تبدیل نمایید و از روی آن DFA کمینه را بسازید.

(۲) پس از اعمال Left-factoring و حذف Left-recursion از گرامر زیر، توابع FIRST و FOLLOW را برای Nonterminal-های آن محاسبه نمایید و از روی آن جدول LL(1) را محاسبه کنید. آیا این گرامر LL(1) است؟ با استفاده از جدول حاصل، ورودی داده شده را با نشان دادن وضعیت پشته و ورودی تجزیه نمایید.

(۳) مجموعه‌ی Itemset-های SLR(1) (یا LR(1) یا LALR(1)) را برای گرامر زیر بسازید و از روی آن جدول‌های ACTION و GOTO را برای تجزیه‌ی^۱ Shift-reduce بسازید. آیا این گرامر LL(1) است؟ این گرامر را به شکلی تغییر دهید که اولویت + در آن کمتر از * باشد. با استفاده از جدول حاصل، ورودی داده شده را با نشان دادن وضعیت پشته و ورودی تجزیه نمایید.

۴) برای گرامر زیر، itemset-ها و جدول‌های GOTO و ACTION را برای الگوریتم LR(1) محاسبه نمایید. سپس با استفاده از این دو جدول، رشته‌ی «aa+aa*+» را با نشان دادن وضعیت پشته، ورودی و عمل ACTION در هر گام، تجزیه^۱ نمایید.

$S \rightarrow S S +$
 $S \rightarrow S S *$
 $S \rightarrow a$

دقت کنید که در این سؤال باید ابتدا گرامر را Augment نمایید (قاعده‌ی $S' \rightarrow S$ را اضافه نمایید). سپس دقت کنید که در اضافه کردن Itemset-ها برای محاسبه‌ی Closure، باید مقدار Lookahead را نیز در نظر بگیرید. برای مثال، Itemset اول برابر است با:

$S' \rightarrow .S$	\$
$S \rightarrow .S S +$	\$
$S \rightarrow .S S *$	\$
$S \rightarrow .a$	\$
$S \rightarrow .S S +$	a
$S \rightarrow .S S *$	a
$S \rightarrow .a$	a

دلیل اضافه شدن سه Item نهایی، قرار گرفتن نقطه قبل از S در Item-های دوم و سوم است که آنچه بعد از آنها قرار می‌گیرد قطعاً با FIRST(S) شروع می‌شود. بنابراین مقدار Lookahead در این سه Item برابر a می‌باشد. این Item را می‌توان به صورت خلاصه‌تر به صورت زیر نشان داد:

$S' \rightarrow .S$	\$
$S \rightarrow .S S +$	\$, a
$S \rightarrow .S S *$	\$, a
$S \rightarrow .a$	\$, a

۵) عبارت منظمی برای comment در زبان C (به شکل `/* comment */`) ارائه دهید.

باید در این تمرین دقت کنید که آنچه در بین «/» و «*/» قرار می‌گیرد نباید شامل «*/» باشد. بنابراین عبارت منظم «/.**/» جواب درستی برای این مسئله نیست. در صورتی که علامت «*» در این عبارت منظم ظاهر شود، نباید بعد از آن علامت «/» قرار بگیرد: مثل عبارت منظم «/.**/» در این عبارت منظم، «[xyz^]» با هر حرفی به غیر از حروف «x»، «y» و «z» منطبق می‌شود.