



تکلیف دوم کامپایلر

پاییز ۹۸

۱- در مورد گرامر زیر، به سوالاتی که در ادامه آمده است پاسخ دهید.

1. $Z \rightarrow S$
 $S \rightarrow E B \mid T$
 $E \rightarrow 1 A B$
 $A \rightarrow 0 A \mid \epsilon$
 $T \rightarrow A b$
 $B \rightarrow b E \mid c E \mid T$

String: 10c1b00b

(الف) یک اشتقاق چپ برای رشته‌ی ورودی بنویسید.

(ب) یک اشتقاق راست برای رشته‌ی ورودی بنویسید.

(ج) یک درخت تجزیه برای این رشته بنویسید.

(د) آیا گرامر مبهم یا غیر مبهم است؟ توضیح دهید.

۲- با توجه به گرامر زیر، کدام عبارت را می‌توان نتیجه گرفت.

- $E \rightarrow E * F \mid F + E \mid F / F \mid E + E$
 $F \rightarrow F - F \mid F - id \mid id - F \mid id - id$

(الف) ضرب اولویت بیشتری نسبت به دیگر عملگرها دارد.

(ب) تفریق اولویت بیشتری نسبت به دیگر عملگرها دارد.

(ج) + و - هر دو دارای اولویت یکسان هستند.

(د) + اولویت بیشتری دارد.

(ه) هیچ کدام.

(و) بسته به رشته‌ی مورد نظر ممکن است هر کدام اولویت بیشتری داشته باشد.

۳- گرامر زیر را برای تجزیه‌ی بالا به پایین آماده کنید و رشته‌ی داده شده را به کمک گرامر به دست آمده تجزیه کنید.

- $E \rightarrow F \mid T$
 $F \rightarrow S + T$
 $S \rightarrow E \mid P$
 $T \rightarrow T B \mid B$
 $B \rightarrow B * \mid P$
 $P \rightarrow a \mid b$
String: $a * b * a a * b$

۴- در گرامرهای مستقل از متن ممکن است به صورت ضمنی بازگشتی چپ وجود داشته باشد. مانند گرامر زیر که

برای متغیر S بازگشتی چپ ضمنی وجود دارد.

- 1) $S \rightarrow A a \mid b$
 $A \rightarrow A d \mid S g$

تحقیق کنید با چه روشی می توان بازگشتی چپ ضمنی را حذف کرد. سپس در گرامر بالا و هر یک از گرامرهای زیر بازگشتی چپ ضمنی را حذف کنید. آیا گرامر حاصل $LL(1)$ می شود؟

$$2) \begin{aligned} A &\rightarrow Ba | Aa | c \\ B &\rightarrow Bb | Ab | d \end{aligned}$$

$$3) \begin{aligned} S &\rightarrow Aa | b \\ A &\rightarrow Ac | Sd \end{aligned}$$

$$4) \begin{aligned} S &\rightarrow AbB \\ A &\rightarrow Sa | cB \\ B &\rightarrow dB | e \end{aligned}$$

$$5) \begin{aligned} \text{Stmt} &\rightarrow \text{if expr then stmt} \quad | \quad \text{matchedStmt} \\ \text{matchedStmt} &\rightarrow \text{if expr then matchedStmt else stmt} \quad | \quad \text{other} \end{aligned}$$

۵- نشان دهید گرامری که $LL(1)$ است نمی تواند مبهم باشد و برعکس، گرامری که مبهم باشد نمی تواند $LL(1)$ باشد.

۶- اگر گرامری $LL(1)$ نباشد، با چه روش هایی می توان برای آن یک گرامر $LL(1)$ معادل ایجاد کرد؟ آیا همواره می توان برای یک گرامر مستقل از متن، گرامر $LL(1)$ معادل ایجاد کرد؟

۷- بررسی کنید هر یک از گرامرهای زیر $LL(1)$ هست یا خیر. اگر گرامری $LL(1)$ نیست، تلاش کنید برای آن یک گرامر $LL(1)$ معادل به دست آورید.

$$1) \begin{aligned} S &\rightarrow ABCSDE | f \\ A &\rightarrow aA | B \\ B &\rightarrow bB | \epsilon \\ C &\rightarrow cC | d \\ D &\rightarrow dD | Be \\ E &\rightarrow gE | \epsilon \end{aligned}$$

$$2) \begin{aligned} S &\rightarrow Aab \\ A &\rightarrow a | \epsilon \end{aligned}$$

$$3) S \rightarrow aS | Sa | g$$

$$4) E \rightarrow E-E | E/E | id$$

۸- برای گرامرهای زیر با رسم جدول تجزیه $LL(1)$ ، بررسی کنید که آیا گرامر $LL(1)$ است؟

$$1) \begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' | \epsilon \\ T &\rightarrow FT' \end{aligned}$$

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

$$F \rightarrow \epsilon \mid id \mid \epsilon$$

$$2) \quad S \rightarrow AB$$

$$A \rightarrow aA \mid C$$

$$B \rightarrow bB \mid \epsilon$$

$$C \rightarrow cC \mid \epsilon$$

۹- رشته‌ی زیر را با توجه به گرامر داده شده و با بهره‌گیری از روش کنترل خطای panic mode تجزیه کنید.

$$E \rightarrow TX$$

$$X \rightarrow +E \mid \epsilon$$

$$T \rightarrow id \mid Y \mid (E)$$

$$Y \rightarrow *T \mid \epsilon$$

String: $id * ((id * + id) + id + id$

پیاده سازی:

۱۰- با یک زبان برنامه نویسی دلخواه برای هر یک از گرامرهای زیر یک تجزیه کننده بازگشتی-کاهشی و یک تجزیه کننده‌ی LL1 (روش جدولی) بنویسید و این دو روش را با هم مقایسه کنید. (در صورت نیاز ابتدا گرامر را LL(1) کنید)

$$1) \quad S \rightarrow AcB$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow bBS \mid \epsilon$$