جلسه هشتم

ساده سازی گرامر های منظم

حذف λ از زبان:

برای ما مهم است که بتوانیم λ را از زبان ها حذف کنیم و به زبان های مستقل از متنی بپردازیم که فاقد λ باشند.اگر G گرامری باشد که $L-\lambda$ را تولید کند، آنگاه افزودن قانون $S_0 \to S \mid \lambda$ و در نظر گرفتن s_0 به عنوان وضعیت شروع، λ نیز تولید خواهد شد.

مثال: λ را از گرامر $s \to asb \mid \lambda$ حذف کنید. جواب $\begin{array}{l} S_0 \to S \mid \lambda \\ S \to aSb \mid ab \end{array}$

قانون جایگزینی: مثال $\begin{array}{l} A \to a \mid aaA \mid ababbAC \mid abbC \\ B \to abbA \mid b \end{array} \Rightarrow \begin{array}{l} A \to a \mid aaA \mid abBC \\ B \to abbA \mid b \end{array}$

حذف قوانین بی فایده: به دو علت یک قانون می تواند بی فایده باشد، اولا اینکه منجر به تولید رشته ای در زبان نشود، ثانیا اینکه از وضعیت شروع قابل دسترسی نباشد. در هر یک از این دو حالت با حذف قانون مربوطه گرامر ساده تر می شود.

مثال. در گرامر $\begin{array}{l} S \to aSb \mid \lambda \mid A \\ A \to aA \end{array}$ با حرکت از S به طرف A و با استفاده از قانون $A \to aA$ هیچ رشته ای تولید نخواهد شد یعنی بودن و نبودن متغیر A و قانون مربوطه اش هیچ تاثیری روی زبان ندارد و میتوان آنهارا حذف کرد.

مثال.در گرامر $\begin{array}{l} S \to A \\ A \to aA \mid \lambda \\ B \to bA \end{array}$ هیچ مسیری از وضعیت شروع به متغیر B وجود ندارد، لذا بودن و نبودن قانون مربوطه اش هیچ تاثیری بر روی زبان ندارد و می توان آن را حذف کرد.

حذف قوانین یکه(واحد):

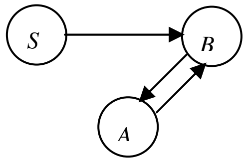
هر قانونی به شکل $\begin{array}{l} A \to B \\ A, B \in V \end{array}$ یک قانون یکه است برا ی این کار ابتدا قوانین یکه و غیر یکه را جداگانه لیست می کنیم سپس گراف وابستگی را با تمامی حالاتی که در قواعد واحد ظاهر شده اند می سازیم، سپس به ازای هر جفت متغیر آنهائی را لیست می کنیم که مسیری بین آنها وجود دارد، سپس با استفاده از قانون جایگذاری قانون های معادلی می نویسیم که قوانین یکه در آنها وجود نداشته باشد، سپس قوانین جدید ایجاد شده را به همراه قوانین غیر یکه اولیه می نویسیم و مطمئن هستیم که گرامر جدید ایجاد شده، همان گرامر اولیه است با این تفاوت که قانون یکه ندارد.

مثال. قوانین یکه گرامر $\begin{array}{l} S \to Aa \mid B \\ B \to A \mid bb \\ A \to a \mid bc \mid B \end{array}$ را حذف کنید.

☐ ابتدا قوانین غیر یکه و یکه را جداگانه لیست می کنیم

قوانین واحد	قوانین غیر واحد
$S \to B$	$S \to aA$
$B \to A$	$B \to bb$
$A \to B$	$A \to a \mid bc$

☐گراف وابستگی را برای متغیر هائی که در قواعد واحد ظاهر شده اند رسم می کنیم



با توجه به گراف $\begin{array}{l} S \to B \Rightarrow S \to bb \\ S \to A \Rightarrow S \to a \mid bc \\ A \to B \Rightarrow A \to bb \\ B \to A \Rightarrow B \to a \mid bc \end{array}$ را بدست می آوریم .

☐ با ترکیب گزینه 1 و 2 جواب نهائی به شکل مقابل خواهد بود. دقت شود که چون از حالت شروع به متغیر B و قوانین مربوطه اش دسترسی نداریم، بود و نبود آن تاثیری ندارد و می توان حذف کرد.

$$\xrightarrow{1,2} \begin{cases} S \to Aa \mid bb \mid a \mid bc \\ B \to bb \mid a \mid bc \quad \times \\ A \to a \mid bc \mid bb \end{cases}$$

حذف بازگشتی چپ:

هر قانونی به شکل $A \to Aa$ را قانون بازگشتی چپ گویند. که در آن $A \in V \ , \ a \in T$

برای حذف قانون بازگشتی چپ که معمولا به شکل $A \to Aa|b$ دیر ه می شود از ایده زیر استفاده می کنیم :

با توجه به عبارت $A \to Aa \to Aaa \to Aaaa \ldots \Rightarrow baaa \ldots a$ ملاحظه می شود که گرامر $A \to Aa|b$ رشته هائی را تولید می کند که یک b اولشان هست و به دنبال آن به تعداد نامشخص a هست پس کافی است که گرامری پیدا کنیم که همان رشته ها را تولید کند و بازگشتی چپ نباشد.گرامر $\begin{matrix} A \to bT \\ T \to aT \end{matrix}$ نتیجه مطلوب را به ما می دهد.

گرامر های نرمال

الف.فرم نرمال چامسکی

گرامری در این فرم است که همه قواعد آن به یکی از دو صورت $A \to BC \ (A,B,C \in V)$ یا $A \to a \ (A \in V \ , \ a \in T)$ باشد(یا ترکیبی از هر دو)

ب.فرم نرمال گریباخ

گرامری در این فرم است که همه قواعد آن به شکل $A \to aV^*$ باشد.

مثال. گرامر $\begin{matrix} S \to ABa \\ A \to aaB \\ B \to Ac \end{matrix}$ را به فرم نرمال چامسکی ببرید. جواب

$$A \to ABa \Rightarrow \begin{cases} S \to AT_1 \\ T_1 \to BT_2 \\ T_2 \to a \end{cases} \quad A \to aab \Rightarrow \begin{cases} A \to T_2T_3 \\ T_3 \to T_2T_4 \\ T_4 \to b \end{cases} \quad B \to Ac \Rightarrow \begin{cases} B \to AT_5 \\ T_5 \to c \end{cases}$$

مثال. گرامر $\begin{matrix} S \to abAB \\ A \to bAB|\lambda \\ B \to BAa|A|\lambda \end{matrix}$ را به فرم نرمال چامسکی ببرید. جواب ابتدا قاعده λ و قاعده یکه را حذف می کنیم .

$$\begin{matrix} S \to abAB|abB|abA|ab \\ A \to bAB|bB|bA|b \\ B \to BAa|Ba|Aa|a \\ B \to bAB|bB|bA|b \end{matrix}$$

$$S \to abAB \Rightarrow \begin{cases} S \to T_1T_2 \\ T_1 \to a \\ T_2 \to T_3T_4 \\ T_3 = b \\ T_4 \to AB \end{cases} \quad S \to abB \Rightarrow \begin{cases} S \to T_1T_5 \\ T_5 \to T_3B \end{cases} \quad S \to abA \Rightarrow \begin{cases} S \to T_1T_6 \\ T_6 \to T_3A \end{cases} \quad S \to ab \Rightarrow \{S \to T_1T_3$$

برای بقیه موارد(A و B) نیز مشابه S عمل می کنیم.

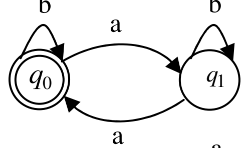
مثال. گرامر $S \to abSb|aa$ را به فرم نرمال گریباخ ببرید. حل :

$$S \to abSb \Rightarrow \begin{cases} S \to aT_1 \\ T_1 \to bST_2 \\ T_2 \to b \end{cases} \quad S \to aa \Rightarrow \begin{cases} S \to aT_3 \\ T_3 \to a \end{cases}$$

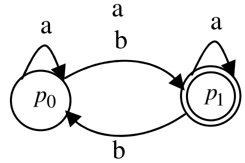
یک مثال خوب در مورد زبان های منظم:

یک NFA رسم کنید که زبان $L = \{w | w \in (a|b)^*, N_a(w) = 2k \ , \ N_b(w) = 2k+1\}$ را بپذیرد. برای حل این مسئله

ابتدا NFA مربوط به $L_1 = \{w | w \in (a|b)^* | N_a(w) = 2k\}$ را رسم می کنیم

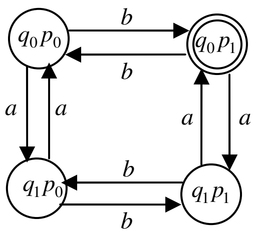


سپس NFA مربوط به $L_2 = \{w | w \in (a|b)^* | N_b(w) = 2k+1\}$ را رسم می کنیم



هر گاه DFA مربوط به L_1 و L_2 را داشته باشیم و بخواهیم یک DFA بکشیم که اشتراک این دو زبان را بپذیرد و مجموعه $\{q_0 \ldots q_n\}$ وضعیت های L_1 باشند، و $\{p_0 \ldots p_n\}$ مجموعه وضعیت های L_2 باشند. یک DFA جدید رسم می کنیم که وضعیت های آن حاصل ضرب دکارتی مجموعه وضعیت های L_1, L_2 است، و داریم $\delta((q_i, p_j), a) = \{\delta(q_i, a), \delta(q_j, a)\}$

در DFA حاصل، وضعیت (q_i, q_j) وضعیت نهائی است اگر و فقط اگر هر دوی q_i و q_j در DFA خودشان وضعیت نهائی باشند. اگر می خواستیم اجتماع دو زبان را بدست آوریم همین کار انجام می شد ولی در انتخاب وضعیت نهائی اگر یکی از وضعیت ها در DFA خودش وضعیت نهائی بود، وضعیت متناظرش را نهائی می کردیم. خلاصه جواب به شکل زیر خواهد بود.



تمرینات اضافی

1- در گرامر زیر متغیر λ را حذف کنید.

$$G:\begin{cases} S \to ABb \mid AB \\ A \to aA \mid \lambda \\ B \to bB \mid \lambda \end{cases}$$

حل: متغیر های A و B به طور مستقیم متغیر λ می باشند و متغیر S به طور غیر مستقیم متغیر λ می باشد. تمام شبه جمله هائی که حداقل یکی از اعضای آن جزء متغیر λ باشد، را داخل جدول قرار می دهیم و تمام جایگشت هائی که این متغیر ها نسبت به λ دارند را به دست می آوریم و قوانین را با درنظر گرفتن این جایگشت ها تولید می کنیم.

S: ABb	
$\lambda\lambda-$	$S \to b$
$-\lambda-$	$S \to Ab$
$\lambda--$	$S \to Bb$
$---$	$S \to ABb$

S: AB	
$\lambda\lambda$	$S \to \lambda$
$-\lambda$	$S \to A$
$\lambda-$	$S \to B$
$--$	$S \to AB$

A: aA	
$-\lambda$	$A \to a$
$--$	$A \to aA$

B: bB	
$-\lambda$	$B \to b$
$--$	$B \to bB$

در پایان تمامی قوانین را جایگزین می کنیم و در آخر نیز تمام λ ها را حذف می کنیم به غیر از $S \to \lambda$ چرا که λ توسط زبان تولید می شود.

$$G' = \begin{cases} S \to ABb \mid Bb \mid Ab \mid b \mid AB \mid B \mid A \mid \lambda \\ A \to aA \mid a \\ B \to bB \mid b \end{cases}$$

2- در گرامر $G:\begin{cases} S \to abA \mid b \mid A \mid B \\ A \to aA \mid b \mid B \\ B \to S \mid Aa \mid bB \end{cases}$ قواعد یکه(واحد)را حذف کنید.

ابتدا قوانین غیر یکه و یکه را جداگانه لیست می کنیم .

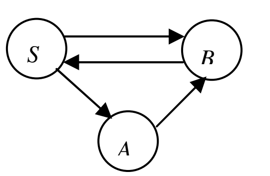
قوانین واحد

$S \to A$
$S \to B$
$A \to B$
$B \to S$

قوانین غیر واحد

$S \to abA \mid b$
$A \to aA \mid b$
$B \to Aa \mid bB$

گراف وابستگی را برای متغیر هائی که در قواعد واحد ظاهر شده اند رسم می کنیم .



اثرات قواعد واحد از روی مسیر های موجود در گراف وابستگی به دست می آید. این قواعد به همراه قواعد غیر واحد گرامر جدید را می سازد

$$S \to A \Rightarrow S \to aA \mid b$$
$$S \to B \Rightarrow S \to Aa \mid bB$$
$$A \to S \Rightarrow A \to abA \mid b$$
$$A \to B \Rightarrow A \to Aa \mid bB$$
$$B \to A \Rightarrow B \to aA \mid b$$
$$B \to S \Rightarrow B \to abA \mid b$$

☐ گرامر جدید به شکل روبرو خواهد بود.

$$S \to abA \mid b \mid aA \mid Aa \mid bB$$
$$A \to aA \mid b \mid abA \mid Aa \mid bB$$
$$B \to Aa \mid bB \mid abA \mid b \mid aA$$

3- متغیر های غیر مفید را در گرامر G پیدا کرده و حذف کنید.

$$G: \begin{cases} S \to aB \mid b \mid Db \\ A \to aB \mid b \\ B \to bB \mid b \\ E \to eE \mid e \\ F \to fF \mid f \\ D \to dD \\ M \to EFe \end{cases}$$

جواب: دقت شود که متغیر D به الفباء نمی رسد(حذف می کنیم) و به بقیه متغیر ها به جزء A از حالت شروع دسترسی نداریم.

$$G' = \begin{cases} S \to aB \mid b \\ B \to bB \mid b \end{cases}$$

4- فرم نرمال گریباخ گرامر $S \to aSa \mid bSb \mid \lambda$ را بنویسید.

گام اول : حذف نماد آغازین بازگشتی

$$S' \to S$$
$$S \to aSa \mid bSb$$
$$S \to \lambda$$

گام دوم : حذف قانون λ

$$S' \to S \mid \lambda$$
$$S \to aSa \mid bSb \mid aa \mid bb$$

گام سوم : نوشتن به فرم نهائی گریباخ

$$S \to aSA \mid bSB \mid aA \mid bB$$
$$A \to a$$
$$B \to b$$

5- فرم نرمال چامسکی گرامر G را بنویسید.

$$G: \begin{cases} S \to aSb \mid aDb \mid \\ D \to aDa \mid bDb \mid \lambda \end{cases}$$

گام اول : حذف قانون λ

$$S \to aSb \mid aDb \mid ab$$
$$D \to aDa \mid bDb \mid aa \mid bb$$

گام دوم :

$$S \to ASb \mid ADB \mid AB$$
$$D \to ADA \mid BDB \mid AA \mid BB$$
$$A \to a$$
$$B \to b$$

گام سوم : فرم نهائی نرمال چامسکی

$$S \to AX \mid AY \mid AB$$
$$X \to SB$$
$$Y \to DB$$
$$D \to AZ \mid BT \mid AA \mid BB$$
$$Z \to DA$$
$$T \to DB$$
$$A \to a$$