



به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر دانشگاه تهران

پروژه ۲ درس نظریه زبان ها و ماشین ها ۹۸ – ۹۷

دستیاران : سعید عباسی – سینا کچوئی

abbasi.sa77@yahoo.com , sina95kachoei@gmail.com

زمان آپلود : ۷ تیر

زمان تحویل حضوری اعلام می گردد

در این پروژه قرار است یک گرامر مستقل از متن دلخواه گرفته و الگوریتم پارس کردن برای آن بنویسید. بدین منظور سه بخش زیر را به ترتیب پیاده سازی خواهید کرد:

بخش اول: ساده سازی گرامر مستقل از متن

۱. حذف قواعد لامبدا

sample input	sample output
$A \rightarrow aA \mid \lambda$ $S \rightarrow ABC$ $A \rightarrow Aa \mid a$ $B \rightarrow Bb \mid \lambda$ $C \rightarrow Cc \mid \lambda$	$A \rightarrow aA \mid a$ $S \rightarrow A \mid AB \mid AC \mid ABC$ $A \rightarrow Aa \mid a$ $B \rightarrow Bb \mid b$ $C \rightarrow Cc \mid c$

۱۱. حذف قواعد یکه

$A \rightarrow aA \mid B$ $B \rightarrow bB \mid AA$	$A \rightarrow aA \mid bB \mid AA$ $B \rightarrow bB \mid AA$
$S \rightarrow A$ $A \rightarrow B$ $B \rightarrow C$ $C \rightarrow S \mid cB \mid c$	$S \rightarrow cB \mid c$ $A \rightarrow cB \mid c$ $B \rightarrow cB \mid c$ $C \rightarrow cB \mid c$

۱۱۱. حذف قواعد بی فایده

i. حذف قواعد غیر مولد

ii. حذف قواعد دسترس ناپذیر از متغیر شروع

$A \rightarrow aA \mid a \mid BA$ $S \rightarrow sAS$ $A \rightarrow aS \mid B$ $B \rightarrow cC \mid c$	$A \rightarrow aA \mid a$
$S \rightarrow aSb \mid ab$ $T \rightarrow tT \mid t$	$S \rightarrow aSb \mid ab$

توجه: توصیه می شود ترتیب ساده سازی بالا را رعایت کنید.

بخش دوم: تبدیل به فرم نرمال چامسکی

قواعد تولید گرامر به فرم نرمال چامسکی به صورت زیر هستند:

$$V \rightarrow V_1V_2$$
$$V \rightarrow \sigma$$

$$V_1, V_2 \in NT$$
$$\sigma \in T$$

در این بخش شما باید گرامر ساده شده ی قسمت قبل را به فرم بالا در آورید.

sample input	...	sample output
$V \rightarrow \sigma_1V_2V_3\sigma_4\sigma_5V_6 \dots$ $V_2 \rightarrow \dots$ $V_3 \rightarrow \dots$ $V_6 \rightarrow \dots$	$V \rightarrow V_1V_2V_3V_4V_5V_6 \dots$ $V_1 \rightarrow \sigma_1$ $V_4 \rightarrow \sigma_4$ $V_5 \rightarrow \sigma_5$ $V_2 \rightarrow \dots$ $V_3 \rightarrow \dots$ $V_6 \rightarrow \dots$	$V \rightarrow V_1T_1$ $T_1 \rightarrow V_2T_2$ $T_2 \rightarrow V_3T_3$ $T_3 \rightarrow V_4T_4$ $T_4 \rightarrow V_5T_5$ $T_5 \rightarrow V_6 \dots$ $V_1 \rightarrow \sigma_1$ $V_4 \rightarrow \sigma_4$ $V_5 \rightarrow \sigma_5$ $V_2 \rightarrow \dots$ $V_3 \rightarrow \dots$ $V_6 \rightarrow \dots$

بخش سوم: پیاده سازی الگوریتم CYK

الگوریتم CYK یک گرامر به فرم نرمال چامسکی به همراه یک رشته گرفته و معین می کند که رشته توسط این گرامر پذیرفته می شود یا خیر.

این الگوریتم به ازای هر رشته n حرفی یک ماتریس $M_{n \times n}$ ساخته و درایه های قطر اصلی و بالای آن را با مجموعه هایی از متغیر های گرامر پر می کند. در نهایت رشته پذیرفته است اگر و تنها اگر متغیر شروع در درایه $M_{1,n}$ یافت شود.

parsing grammar	with word	results in
$S \rightarrow AB$ $A \rightarrow BB \mid a$ $B \rightarrow AB \mid b$	$aabbb$ ab ba $aabbabaaab$	$accept$ $accept$ $reject$ $reject$
$A \rightarrow AB \mid CD$ $B \rightarrow AB \mid CD$ $C \rightarrow AB \mid CD$ $D \rightarrow AB \mid CD$ $S \rightarrow AB \mid CD$ $A \rightarrow x$ $B \rightarrow x$ $C \rightarrow x$ $D \rightarrow x$	x xx xxx	$reject$ $accept$ $accept$

توضیح: برای گرامر اول و رشته 'aabbb' ماتریس تولید شده توسط الگوریتم به صورت زیر خواهد بود:

A		S, B	A	S, B
	A	S, B	A	S, B
		B	A	S, B
			B	A
				B

توضیح ورودی و خروجی:

ابتدا قواعد گرامر را خط به خط خوانده تا به علامت \$ در یک خط که نشان دهنده اتمام گرامر است برسید.

حال اگر گرامر در فرم نرمال چامسکی بود $True$ در غیر این صورت $False$ چاپ کنید و در خطوط بعد گرامر معادل آن به فرم نرمال چامسکی را چاپ کنید.

سپس در هر خط یک رشته می خوانید و به ازای آن خروجی $True$ یا $False$ متناسب با پذیرفته شدن یا نشدن رشته مورد نظر روی گرامر می دهید تا باز به علامت \$ که نشان دهنده اتمام رشته هاست برسید.

توضیحات تکمیلی:

- در تعریف قواعد تولید گرامر ورودی متغیر ها حروف انگلیسی بزرگ و ترمینال ها حروف انگلیسی کوچک هستند.
- متغیر شروع گرامر ورودی همواره **S** است.
- زبان گرامر های ورودی شامل λ نیست. (قاعده $\lambda \rightarrow \dots \rightarrow \dots$ نداریم)
- تابعی برای چاپ گرامر داشته باشید تا در صورت نیاز هنگام تحویل پس از هریک از بخش های یک و دو آن را فراخوانی کنید.
- تابعی برای نمایش ماتریس حاصل از اجرای الگوریتم CYK داخل کد داشته باشید تا در صورت نیاز هنگام تحویل فراخوانی کنید.
- کد خود را در یک فایل zip به فرمت SID.zip که SID شماره دانشجویی شماست آپلود کنید.
- موارد مطروحه در مورد **CA** را با دستیاران آموزشی مربوطه مطرح کنید.
- سعی کنید خودتان برای تایید صحت عملکرد برنامه، تعدادی ورودی خروجی نمونه که حالات مختلف را پوشش دهد ایجاد نمایید.