



گزارش پروژه درس نظریه زبان ها و ماشین ها

نام استاد: رضا انتظاری

نام دانشجو: محمدمهدی جاوید

سوال یک :

Grammar

ابتدا الگوریتم را به CNF تبدیل کرده و سپس با استفاده از الگوریتم CYK پذیرش یا عدم پذیرش آن را بررسی می کنیم.

مراحل برای تبدیل کردن به CNF (Chomsky Normal Form) :

۱- ابتدا یک استیت S_0 اضافه کرده (فرض شده که S استارت استیت قبلی بوده)

$\langle S_0 \rangle \rightarrow \langle S \rangle .a$

۲- سپس به حذف Useless Productions می پردازیم

a. به جای ترمینال ها متغیرهایی که وجود دارد یا متغیرهایی جدید تولید کرده و جایگذاری

می کنیم.

3- $\langle A \rangle \rightarrow a\langle A \rangle$

4- $\langle B \rangle \rightarrow a$

5- $\langle A \rangle \rightarrow \langle B \rangle \langle A \rangle$ (جایگزین)

۳- برای تبدیل کردن به فرم چامسکی تنها وجود دو متغیر یا تنها وجود یک ترمینال یا رفتن متغیر

شروع به لامبدا امکان پذیر است امکان پذیر است

$\langle S \rangle \rightarrow \langle A \rangle \langle B \rangle .a$

$\langle A \rangle \rightarrow a .b$

$\langle S_0 \rangle \rightarrow \# .c$

۴- حال باید با تبدیل سمت راست متغیرها به دو متغیر یا وجود یک ترمینال این کار را به اتمام رساند

a. در هر مرحله یک متغیر از سمت راست آن‌ها کم خواهد شد
i. دوتا دوتا متغیر از سمت راست برداشت و به جای آن یک متغیر جدید تعریف می‌کنیم

ii. $\langle S \rangle \rightarrow \langle A \rangle \langle B \rangle \langle C \rangle$

iii. $\langle S \rangle \rightarrow \langle A \rangle \langle D \rangle$

iv. $\langle D \rangle \rightarrow \langle B \rangle \langle C \rangle$

v. سه متغیر داشتیم که در یک مرحله یک متغیر کم شد

۵- نال پروداکشن‌ها یا لامبداها را پاک می‌کنیم.

a. متغیر A به لامبدا می‌رود

b. $\langle S \rangle \rightarrow \langle B \rangle \langle A \rangle$

c. $\langle A \rangle \rightarrow \#$

d. $\langle S \rangle \rightarrow \langle B \rangle \langle A \rangle \mid \langle B \rangle$

e. همانطوری که در مثال بالا می‌بینیم به سمت راست متغیرهایی که به متغیر A خواهند رفت بدون متغیر A را به پروداکشن‌ها اضافه خواهیم کرد.

۶- تمامی یونیت پروداکشن‌ها (Unit Productions) را حذف می‌کنیم

a. اگر یک متغیر تنها به متغیر دیگری برود به طوری که جز آن متغیر چیز دیگری نباشد می‌توان تمام پروداکشن‌های متغیر سمت راست را به متغیر سمت چپ اضافه کند.

b. $\langle S \rangle \rightarrow \langle A \rangle$

c. $\langle A \rangle \rightarrow a \mid b$

d. $\langle S \rangle \rightarrow a \mid b$ جایگزین شده

e. $\langle S \rangle \rightarrow \langle A \rangle$ حذف خواهد شد

مراحل انجام الگوریتم CYK :

- ۱- به طوری که ترکیب همه رشته‌ها را تولید می‌کنیم برای مثال در سطر اول تک کارکتر را بررسی می‌کنیم و متغیرهایی را که به آن می‌روند را در سطر اول نوشته
- ۲- برای سطر دوم کارکترها را دو به دو گرفته و متغیرها را ضرب کارتزین کرده و در صورتی که بتوان آن‌ها را با متغیر دیگری تولید کرد درون مجموعه‌ای ریخته و در هر خانه جدول قرار می‌دهیم
- ۳- همین کار را برای ۳ کارکتر انجام داده و این روند را ادامه می‌دهیم تا به یک خانه برسیم
- ۴- در انتها بررسی می‌کنیم که آیا ریشه یا S0 درون خانه آخر جدول قرار دارد یا خیر
- ۵- در صورت وجود رشته پذیرش می‌شود. و بر عکس
- ۶- در این شکل سطر اول که هر کارکتر رشته هستند وجود پروداکشن آن‌ها بررسی شده و درون خانه‌های جدول قرار داده شده‌اند

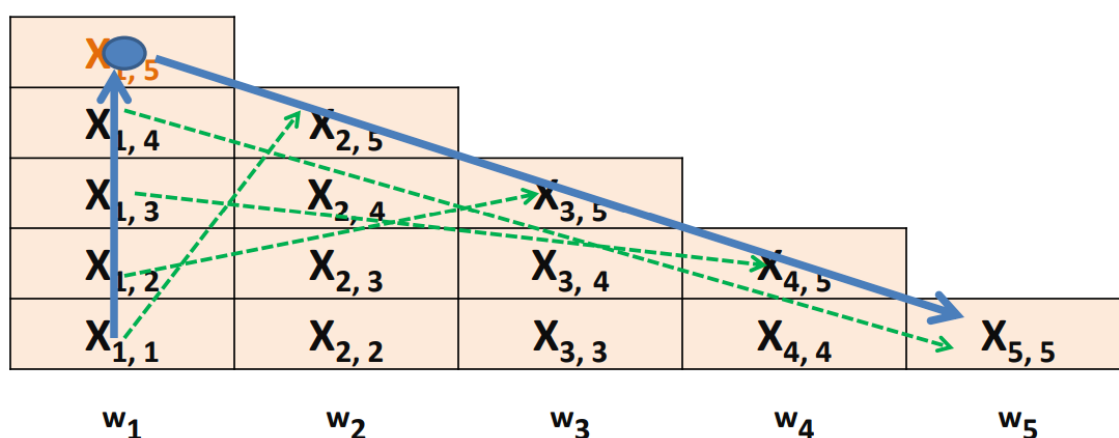
Constructing The Triangular Table

{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

$S \rightarrow AB \mid BC$
 $A \rightarrow BA \mid a$
 $B \rightarrow CC \mid b$
 $C \rightarrow AB \mid a$

Calculating the Bottom ROW

Construct a Triangular Table



۷- همین روند را تا سطر آخر تکرار کرده و پذیرش یا عدم پذیرش آن مشخص خواهد شد.

$X_{i,i}$ is the set of variables A such that

$A \rightarrow w_i$ is a production of G

$(X_{i,i}, X_{i+1,j}), (X_{i,i+1}, X_{i+2,j}) \dots (X_{i,j-1}, X_{j,j})$

Turing Machine

Transitions:

$$\delta(q_1, a) = (q_2, b, L) \quad \delta(q_2, b) = (q_3, c, R)$$

Encoding:

1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1

separator

33

فرمت رشته ورودی به صورت بالاست به طوری که با دو تا صفر ترنزیشن ها از یکدیگر جدا شده اند.

حال هر ترنزیشن را بر اساس یک صفر بین آنها ترجمه می کنیم به طوری که

Current State , Read Character, Next State, Convert Character in the Tape, Move

حال هر استیت چندین ترنزیشن دارد که کارکتری که باید خوانده شود تا به استیت بعدی برود به عنوان کلید دیکشنری است. و استیت بعدی به عنوان مقدار دیکشنری استفاده شده. که دقیقاً با ترتیب بالا این مقادیر ذخیره شده اند.

نکته مهم برای کارکترها این است که کارکتر بلنک به صورت یک عدد یک ذخیره می‌شود.

ابتدا ورودی اول را گرفته و پارس می‌کنیم و ترنزشن‌ها به وجود می‌آیند.

سپس با خواندن کوئری‌ها آن‌ها را پیمایش خواهیم کرد.

از استیت اولیه شروع کرده و به سمت جلو حرکت می‌کنیم. اگر به استیت نهایی رسیدیم کوئری پذیرش می‌شود.

در صورتی که از نوار بیرون بزند فرض شده که از بلنک استفاده می‌شود.

Halt در استیت پایانی : پذیرش و در غیر این صورت عدم پذیرش اتفاق خواهد افتاد.

در زیر نحوه ترجمه tape ورودی را می‌بینیم :

```
11011011 --> 2, 2, 2 --> a, a, a
110111011 --> 2, 3, 2 --> a, b, a
```

در زیر ترجمه رشته ورودی کد شده را خواهیم دید :

```
101101011011 1010110101
sigma(q1, a) = sigma(q1, a, R)
sigma(q1, blank) = sigma(q2, blank, L)
```