



بنام خدا

تجزیه Bottom up

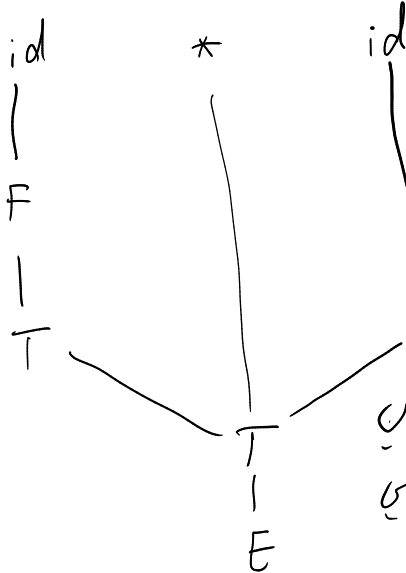
(مثال) $E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$

درودی $id * id$

$E \rightarrow T \rightarrow T * F \rightarrow T * id \rightarrow F * id \rightarrow id * id$ ✓

Left-to-right تجزیه

right most اشتقاق



مراحل تجزیه به صورت BU برعکس مراحل اشتقاق right most تجزیه

چون درخت از بزرگ به ریشه ساخته می شود در از آنجایی که همچنان درودی از صلب به است خوانده می شود اولین ترسیمی که جایگزین می شود است چه درودی قرار داد که در مراحل اشتقاق آخرین سبیل می هستند که تولید شده اند.

ایده تجزیه BU

یک - BU راست ترین اشتقاق را به صورت عکس دنبال می کند

آخر - $\alpha \beta \omega$ یک مرحله از پارس BU باشد ($\alpha \beta \omega$ یک رشته (حرم جمله ای) اشتقاق است)

فرض کنید مرحله بعدی پارس از طریق $X \rightarrow \beta$ انجام می شود یعنی β با X جایگزین می شود

 $X \in N$

$S \xrightarrow{*} \alpha X \omega \rightarrow \alpha \beta \omega$

 $X \rightarrow \beta$

ω رشته ای از ترمینال است

$\alpha \underbrace{\beta}_{\gamma \in T} \omega$

$Y \rightarrow \alpha \gamma a$

$\alpha \beta \omega \rightarrow \alpha X \omega$ مرحله ای در راست ترین اشتقاق است یعنی وقتی

X را جایگزین می کنیم همه سبیل می له باید ترمینال باشند که نسبت اشتقاق به X رسیده باشد یعنی در تجزیه در این مرحله می خواهیم β را با X جایگزین کنیم

سوال چرا nAy را به عنوان β انتخاب کردیم و $nAya$ را انتخاب نکردیم؟
 (با فرض اینکه production می به صورت $nAya \rightarrow$ داریم)



رابطه
 $\alpha \quad \beta \quad \gamma$
 چنانچه

روش برای تقسیم رشته فعلی به دو زیر رشته نیاز داریم

سوال اگر در تجربه رشته ای مثل $\{NUT\}$ $\frac{ET}{\gamma 2}$ رسیده باشیم می توانیم در مرحله بعد n را هم از ورودی بخوانیم یا نباید بخوانیم. پس خطاهای تقسیم رشته همین تصمیم است.

در تجربه BU دو محل متفاوت انجام می شود

۱- **Shift** یک توکن دیگر از ورودی بخوانیم (جدای از به سمت راست حرکت کند) **انتقال**

۲- **reduce** در رشته فعلی (در خرم های مجله ای اشتقاق) یک زیر رشته را که درست است یکی از production های بنیم **کاهش** با غیر ترسینال حرکت به production جایگزین کنیم

مثال) $E \rightarrow T + E \mid T$
 $T \rightarrow id * T \mid id$

ورودی $id * id + id$

① $id * id + id$ **شیت**

② $id * id + id$ **S**

③ $id * id + id$ **S**

④ $id * id + id$ **✓**

⑤ $id * T + id$ **✓**

⑥ $T + id$ **S**

⑦ $T + id$ **S**

$id * id + id$ **شیت**
 $T * id + id$ **reduce**

X اگر از هم به بیست می خوانیم

⑧ $T + id$ **✓**

⑨ $T + T$ **✓**

⑩ $T + E$ **✓**

⑪ E **✓**



به یاد داشته باشید

عملیات جداسازی ورودی به دو زیررشته را می توان از طریق استک پیاده سازی کرد
پسین مرتب که در انتهای استک خالی است و یک رسته ورودی داریم. حوازی می توان از زیررشته ورودی با عمل shift یک بزرگ
به استک اضافه کرد یا زیررشته ای از روی استک برداشت و کاهش داد

حال در سوال پیش می آید
در هر مرحله چه احساس تصمیم بگیریم که shift یا reduce انجام دهیم
shift-reduce conflict
S-r
اگر امکان کاهش از طریق پسین از یک production متفاوت وجود داشته باشد
reduce-reduce conflict
r-r
با کدام یک reduce را انجام دهیم

هنگام handle

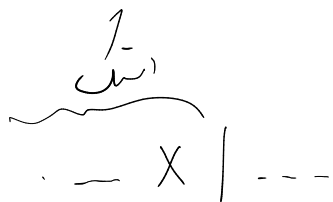
$$S \xRightarrow{*} \alpha X \omega \rightarrow \alpha \beta \omega$$

$$X \rightarrow \beta$$

در عملیات تجزیه اگر به نرم جمله ای $\alpha \beta \omega$ رسیدیم و
ا ω از روی متن خوانده نشده بتوانیم β را به X کاهش دهیم
در این صورت β را هنگام $\omega \alpha \beta$ کوئیم

هنگام زیررشته از متن خالی دو غیر ترسیال است نه از آن را کاهش دهیم، کاهش دی بعد از آن می توانیم بنایا به
S ختم شوند.

ما می خواهیم فقط روی هنگام کاهش دهیم



در تجزیه $\omega \beta$ با استفاده از عمل shift/reduce، هنگامی که فقط روی استک
ظا حرمی شوند و نه درون آن.

- با فاصله بعد از کاهش یک هنگام، یک غیر انتهای روی استک قرار می گیرد (X)
X در اشتقاق، راست ترین غیر انتهای بوده است. هنگام بعدی باید سمت راست X باشد
زیر اشتقاق یک اشتقاق right most بوده است
پس دنباله ای از حرکات شیفت مارا به هنگام بعدی می رساند.

مثال $T \rightarrow (E)$
ا و (E) می تواند روی استک ظاهر شوند. (E) می تواند ظاهر شود
چون (E) هنگامی است که باید قبلاً کاهش داده شود و اگر از X گذر کنیم دیگر هیچ وقت نمی توانیم (E) را بارس کنیم



باز به تعریف مبهم خبریه برای مشخص کردن محتوای محلی درون استک داریم

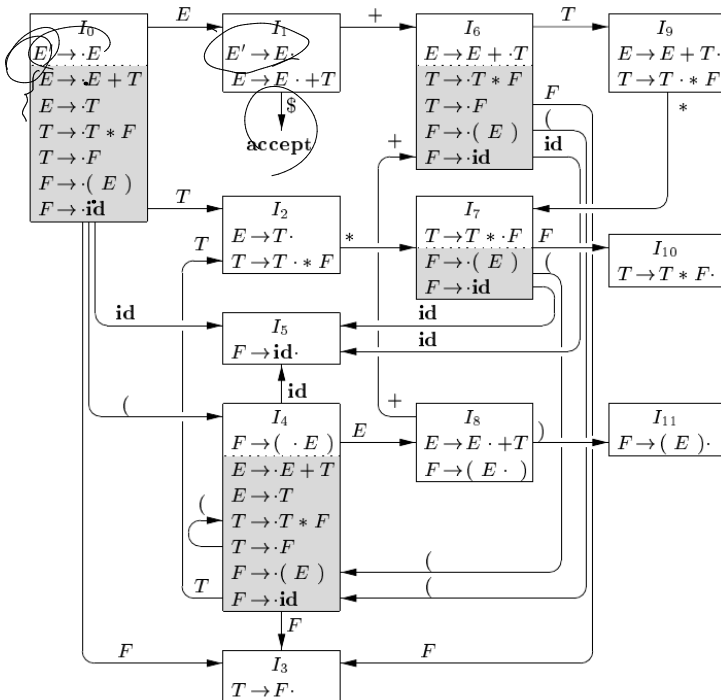
viable prefix عرض کنید در یک مرحله تجزیه $shift/reduce$ حرکت داریم، la داریم، la محتوای درون استک و ra رشته ای از ورودی است که هنوز حرکت نکرده، آنگاه la را **viable prefix** گوئیم

سوال: چه رشته هایی در **viable prefix** باشند؟

اگر در یک مرحله با شیفت کردن بعدی درون استک، محتوای استک تبدیل به رشته ای شود که p نیست آن شیفت مجاز نمی باشد

روش کلی کار: به ازای هر گرامر یک DFA می سازیم که تمام رشته ای که در گرامر این DFA پذیرش می شوند la می آید آن گرامر هستند

مثالی از این DFA



گرامر

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid id \end{aligned}$$

روش ساخت DFA مورد نظر

تعریف آئتم: در تجزیه LR، state ای دنبال می شوند تا به یاد داشته باشیم کجای دست یار هستیم هر state مجموعه ای از آئتم است.

آئتم برای هر production گرامر همان production است که در محلی از به نه آن - اضافه شده است.

مثال $T \rightarrow (E)$

$$\begin{cases} T \rightarrow \cdot (E) \\ T \rightarrow (\cdot E) \\ T \rightarrow (E \cdot) \\ T \rightarrow (E) \cdot \end{cases}$$

به این آئتم، آئتم های LR(a) گوئیم



و تئ داریم $A \rightarrow \cdot X YZ$ یعنی انتظار داریم X را ببینیم و تئ داریم $A \rightarrow X \cdot YZ$ یعنی X را دیده ایم، انتظار داریم YZ را ببینیم

حاصله بعدی کار را به دست آوردن state های DFA که شامل گرانجری هستند، زیرا ساخت DFA