

اصول طراحی کامپایلر

گروه کامپیوتر دانشگاه آزاد اسلامی واحد زنجان

مدرس: مهدی بازرگانی
آخرین بروز رسانی: پاییز ۱۳۹۳

M.Bazargani 93-94-2۱

فصل چهارم

تحلیل نحوی
(بخش دوم)

M.Bazargani 93-94-2۲

تحلیل نحوی

❖ تحلیل گر نحوی و خطاهای نحوی

❖ گرامر و گرامر مستقل از متن

❖ تجزیه بالا به پایین و پایین به بالا

❖ تجزیه پیشگو

❖ گرامرهای LL(1)، LR، SLR و LALR

M.Bazargani 93-94-2۳

فواید گرامرها

۱- نمایش دقیق و قابل فهمی برای زبان

۲- امکان ایجاد پارسرهای کارآمد با قابلیت تشخیص ساختارهای نحوی درست و دقیق

۳- ایجاد ساختاری مناسب برای زبان جهت ترجمه صحیح و آشکارسازی خطا توسط گرامر درست طراحی شده

۴- سادگی اضافه نمودن ساختارهای جدید به زبان

M.Bazargani 93-94-2۴

تجزیه کننده

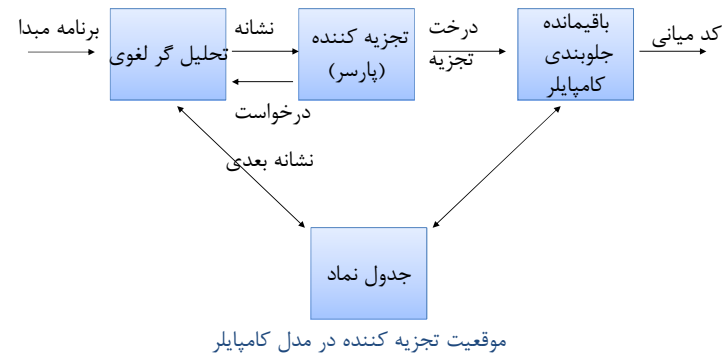
دریافت رشته ای از نشانه ها از تحلیل گر لغوی و بررسی تعلق رشته به زبان توسط گرامر

انجام بررسی طبق ساختارهای نحوی زبان و هر مرحله گزارش خطاهای نحوی به اداره کننده خطا

رفع خطا برای پردازش ادامه ورودی بر اساس خطاهای متداول

M.Bazargani 93-94-2

تجزیه کننده- ارتباطات



M.Bazargani 93-94-2

۶

بررسی روشهای پارسینگ پایین به بالا

یکی از عمومی ترین روشها؛ روش پارسینگ انتقال-کاهش (Shift-Reduce) نام دارد.

روشهای پارسینگ انتقال-کاهش به عنوان یک روش پارسینگ پایین به بالا درخت پارسینگ را از برگها به سمت ریشه تشکیل می دهند.

از رشته ورودی پارسینگ را آغاز نموده و با استفاده از قواعد گرامر سعی می کند رشته ورودی را به عنصر شروع گرامر کاهش دهند.

در هر مرحله از کاهش یک زیررشته ویژه از رشته ورودی که با سمت راست یکی از قواعد منطبق است با نماد سمت چپ آن جایگزین می شود.

ترتیب انتخاب قواعد در پارسینگ پایین به بالا منطبق با ترتیب وارونه ی استفاده از قواعد در اشتقاق سمت راست ترین می باشد.

۷

پارسرهای انتقال کاهش Shift-Reduce Parser

پارسینگ انتقال کاهش به سه نوع متفاوت تقسیم بندی می شود:

پارسینگ تقدم عملگر

پارسینگ تقدم ساده

پارسینگ LR :

SLR
CLR
LALR

M.Bazargani 93-94-2

۸

تجزیه پایین به بالا - انتقال کاهش

سعی در ایجاد درخت تجزیه با شروع از برگها و رفتن به سمت ریشه = کاهش

جایگزینی یک زیررشته خاص منطق با سمت راست یک قانون با نماد سمت چپ همان قانون

M.Bazargani 93-94-2

۹

پارسرهای انتقال کاهش Shift-Reduce Parser

• نکته مهمی که در اینجا وجود دارد انتخاب زیررشته در هر مرحله از انجام کاهش در پارسینگ پایین به بالا است چرا که در برخی از مراحل ممکن است بتوان چند زیررشته را با سمت راست ترین قواعد منطبق نمود.

• چنانچه یک گرامر مبهم نباشد در هر مرحله از کاهش فقط و فقط یک دستگیره وجود دارد اما در صورتی که گرامر مورد نظر مبهم باشد در برخی موارد کاهش ممکن است بیش از یک دستگیره بتوان مشخص کرد.

• حال کدامیک از زیررشته هارا باید برای Reduce انتخاب نمود ؟

M.Bazargani 93-94-2

۱۰

تعریف چند مفهوم (Shift-Reduce)

عبارت (Phrase): بخشی از یک فرم جمله ای است (شامل پایانی و غیر پایانی) که از یک غیرترمینال بوجود آمده باشد.

$$S \Rightarrow \alpha Ax \Rightarrow \alpha \beta \gamma$$

$$S \Rightarrow \alpha Ay \Rightarrow \alpha \beta \gamma$$

عبارت ساده (simple phrase): عبارتی است که در یک مرحله اشتقاق بوجود می آید.

دستگیره (handle): عبارت ساده ای است که در جهت عکس یک اشتقاق سمت راست ترین تولید شده باشد.

$$S \Rightarrow \alpha Ax \Rightarrow \alpha \beta x$$

نکته: سمت راست دستگیره هیچ غیرپایانی وجود ندارد.

M.Bazargani ۹۳-۹۴-۲

۱۱

مثال

با استفاده از گرامر مقابل مراحل پارسینگ پایین به بالا و بالا به پایین رشته abbcdce را مشخص نمایید؟

S → aABe
A → Abc \ b
B → d

دنباله ورودی abbcdce

abbcdce	انتقال
aAbcdce	کاهش
aAde	انتقال
aABe	کاهش
S	

اشتقاق S → aABe → aAde → aAbcdce → abbcdce

M.Bazargani 93-94-2

۱۲

تعریف دستگیره

دستگیره دنباله ایست که منطبق بر سمت راست یک قاعده تولید بوده و کاهش آن به سمت چپ قاعده تولید یک مرحله از مراحل معکوس سمت راست ترین اشتقاق است.

نکته

اگر گرامر مبهم نباشد فقط یک دستگیره در هر مرحله یافت خواهد شد. و اگر گرامر مبهم باشد در هر مرحله چند دستگیره ممکن است یافت شود.

تجزیه کننده های پایین به بالا دو مرحله ی اصلی زیر تکرار می کنند:

پیدا کردن دستگیره

کاهش دستگیره

پارسر انتقال کاهش - دستگیره

زیر رشته ای منطبق بر سمت راست یک قانون و ایجاد کننده یک کاهش به غیر پایانه سمت چپ آن قانون



ویژگی دستگیره

زیر رشته ای که با عمل کاهش با توانایی هدایت تجزیه کننده به عنصر شروع گرامر

مثال دستگیره

دنباله ورودی abbcde

$S \rightarrow aABe$
 $A \rightarrow Abc \setminus b$
 $B \rightarrow d$

abbcde انتقال
aAbcde کاهش
aAde انتقال
aABe کاهش
S

دستگیره ها

$A \rightarrow b$
 $A \rightarrow Abc$
 $B \rightarrow d$

مثال دستگیره

گرامر

مرحله ۱

(1) $E \rightarrow E + E$

(2) $E \rightarrow E * E$

(3) $E \rightarrow (E)$

(4) $E \rightarrow id$

سمت راست ترین اشتقاق

$$\begin{aligned} E &\rightarrow E + E \\ &\rightarrow E + E * E \\ &\rightarrow E + E * id3 \\ &\rightarrow E + id2 * id3 \\ &\rightarrow id1 + id2 * id3 \end{aligned}$$

M.Bazargani 93-94-2۱۷

مثال دستگیره

دستگیره ها

جمله	دستگیره
$id1 + id2 * id3$	id1
$E + id2 * id3$	id2
$E + E * id3$	id3
$E + E * E$	$E * E$

M.Bazargani 93-94-2۱۸

دستگیره- هرس نمودن

مزیت هرس نمودن

توانایی تولید معکوس سمت راست ترین اشتقاق

مراحل هرس نمودن

۱- اگر W جمله گرامر برای تجزیه باشد، آنگاه $W = Y_n$ شبه جمله راست n ام از سمت راست ترین اشتقاق نامشخص .

۲- یافتن دستگیره B_{n-1} در Y_{n-1} و کاهش آن تا بدست آمدن شبه جمله راست Y_{n-2} .

۳- توقف عملیات در صورت رسیدن به عنصر شروع گرامر S

M.Bazargani 93-94-2۱۹

مثال هرس نمودن دستگیره

(1) $E \rightarrow E + E$

(2) $E \rightarrow E * E$

(3) $E \rightarrow (E)$

(4) $E \rightarrow id$

id1 + id2 * id3

قانون کاهشی	دستگیره	شبه جمله راست
$E \rightarrow id$	id1	$id1 + id2 * id3$

M.Bazargani 93-94-2۲۰

مثال هرس نمودن دستگیره

شبه جمله راست	دستگیره	قانون کاهش
$id1 + id2 * id3$	$id1$	$E \rightarrow id$
$E + id2 * id3$	$id2$	$E \rightarrow id$

M.Bazargani ۹۳-۹۴-۲

۲۱

مثال هرس نمودن دستگیره

شبه جمله راست	دستگیره	قانون کاهش
$id1 + id2 * id3$	$id1$	$E \rightarrow id$
$E + id2 * id3$	$id2$	$E \rightarrow id$
$E + E * id3$	$id3$	$E \rightarrow id$

M.Bazargani ۹۳-۹۴-۲

۲۲

مثال هرس نمودن دستگیره

شبه جمله راست	دستگیره	قانون کاهش
$id1 + id2 * id3$	$id1$	$E \rightarrow id$
$E + id2 * id3$	$id2$	$E \rightarrow id$
$E + E * id3$	$id3$	$E \rightarrow id$
$E + E * E$	$E * E$	$E \rightarrow E * E$

M.Bazargani ۹۳-۹۴-۲

۲۳

مثال هرس نمودن دستگیره

شبه جمله راست	دستگیره	قانون کاهش
$id1 + id2 * id3$	$id1$	$E \rightarrow id$
$E + id2 * id3$	$id2$	$E \rightarrow id$
$E + E * id3$	$id3$	$E \rightarrow id$
$E + E * E$	$E * E$	$E \rightarrow E * E$
$E + E$	$E + E$	$E \rightarrow E + E$

M.Bazargani ۹۳-۹۴-۲

۲۴

مثال هرس نمودن دستگیره

قانون کاهشی	دستگیره	شبه جمله راست
$E \rightarrow id$	id1	$id1 + id2 * id3$
$E \rightarrow id$	id2	$E + id2 * id3$
$E \rightarrow id$	id3	$E + E * id3$
$E \rightarrow E * E$	$E * E$	$E + E * E$
$E \rightarrow E + E$	$E + E$	$E + E$
		E

M.Bazargani ۹۳-۹۴-2

۲۵

مشکلات هرس نمودن دستگیره

- ۱- تعیین زیر رشته مناسب برای کاهش در یک شبه جمله راست
- ۲- انتخاب قانون مناسب در موارد وجود دو یا بیشتر قانون با زیر رشته یکسان در سمت راست



M.Bazargani 93-94-2

۲۶

پیاده سازی پارسینگ Shift – Reduce با استفاده از پشته

❖ روش پارسینگ انتقال کاهش نیاز به دو ساختمان داده بنام بافر برای نگهداری رشته ورودی و **stack** برای شناسایی دستگیره و چند عملگر نظیر **shift** و **reduce** نیاز داریم.

❖ در آغاز عمل پارسینگ یک علامت \$ روی پشته و یکی هم در انتهای رشته ورودی قرار داده می شود.

❖ عمل پارسینگ انتقال کاهش به این صورت است که پارسر آنقدر دو عمل انتقال کاهش را انجام می دهد تا اینکه یک خطای نحوی رخ دهد. یا اینکه به حالت نهایی و خاتمه پارسینگ برسد.

❖ حالت خاتمه پارسینگ وقتی رخ می دهد که درون پشته فقط علامت \$ وجود داشته باشد و از رشته ورودی نیز فقط و فقط علامت \$ باقی مانده باشد.

M.Bazargani 93-94-2

۲۷

پیاده سازی پارسینگ Shift – Reduce با استفاده از پشته

استفاده از پشته به منظور نگهداری نمادهای گرامر
استفاده از میانگیر ورودی جهت نگهداری رشته مورد نظر برای تجزیه

۱- انتقال صفر یا چند نماد به پشته توسط تجزیه کننده

۲- ادامه مرحله ۱ تا زمان پیدا شدن یک دستگیره در بالای پشته

۳- کاهش دستگیره پیدا شده به سمت چپ قانون گرامری مناسب آن

روند تجزیه

M.Bazargani ۹۳-۹۴-2

۲۸

پیاده سازی پارسینگ Shift – Reduce با استفاده از پشته

انتقال

در هر مرحله از عمل انتقال یک ترمینال از شروع ورودی به روی پشته منتقل می شود.
عمل انتقال تا زمانی ادامه می یابد که یک دستگیره بالای پشته تشخیص داده شود.

کاهش

در صورتی که یک دستگیره بالای پشته ظاهر شود آنگاه دستگیره از بالای پشته حذف و به جای آن غیرترمینال سمت چپ قاعده ای که سمت راست آن مطابق دستگیره است وارد پشته می شود.

پذیرش

در این حالت پارسر پایان موفقیت آمیز عمل پارسینگ را اعلام می کند.

خطا

پارسر یک خطای نحوی را تشخیص داده و رویه خطاپردازی را فراهم می کند.

M.Bazargani 93-94-2

۲۹

پیاده سازی پارسینگ Shift – Reduce با استفاده از پشته

در حالت کلی با استفاده از انتقال توکن جاری روی استک بر میگذارد، در مرحله کاهش یک دستگیره را از بالای استک برداشته و به جای آن یک غیرپایانی قرار می دهد.
در این روش یک جدول تشکیل می شود و برحسب اینکه چه ورودی، ورودی جاری است، ورودی استک چه عنصری قرار دارد، یکی از اعمال چهارگانه ی بالا انجام می شود.

M.Bazargani 93-94-2

۳۰

مثال تجزیه انتقال کاهش با پشته

گرامر

(1) $E \rightarrow E + E$
(2) $E \rightarrow E * E$
(3) $E \rightarrow (E)$
(4) $E \rightarrow id$

$E \rightarrow E + E$
 $\rightarrow E + E * E$
 $\rightarrow E + E * id3$
 $\rightarrow E + id2 * id3$
 $\rightarrow id1 + id2 * id3$

رشته $id1 + id2 * id3$

M.Bazargani 90-91-2

۳۱

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	$id1 + id2 * id3$ \$	Shift

M.Bazargani 93-94-2

۳۲

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id

M.Bazargani 93-94-2۳۳

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift

M.Bazargani 93-94-2۳۴

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift

M.Bazargani 93-94-2۳۵

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by E → id

M.Bazargani 93-94-2۳۶

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by E → id
\$E + E	* id3 \$	Shift

M.Bazargani 93-94-2۳۷

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by E → id
\$E + E	* id3 \$	Shift
\$E + E *	id3 4	Shift

M.Bazargani 93-94-2۳۸

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by E → id
\$E + E	* id3 \$	Shift
\$E + E *	id3 4	Shift
\$E + E * id3	\$	Reduce by E → id

M.Bazargani 93-94-2۳۹

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by E → id
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by E → id
\$E + E	* id3 \$	Shift
\$E + E *	id3 4	Shift
\$E + E * id3	\$	Reduce by E → id
\$E + E * E	\$	Reduce by E → E * E

M.Bazargani 93-94-2۴۰

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by $E \rightarrow id$
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by $E \rightarrow id$
\$E + E	* id3 \$	Shift
\$E + E *	id3 4	Shift
\$E + E * id3	\$	Reduce by $E \rightarrow id$
\$E + E * E	\$	Reduce by $E \rightarrow E * E$
\$E + E	\$	Reduce $E \rightarrow E + E$

M.Bazargani 93-94-2

۴۱

مثال پارسینگ انتقال کاهش با پشته

پشته	ورودی	عمل
\$	id1 + id2 * id3 \$	Shift
\$id1	+ id2 * id3 \$	Reduce by $E \rightarrow id$
\$E	+ id2 * id3 \$	Shift
\$E +	id2 * id3 \$	Shift
\$E + id2	* id3 \$	Reduce by $E \rightarrow id$
\$E + E	* id3 \$	Shift
\$E + E *	id3 4	Shift
\$E + E * id3	\$	Reduce by $E \rightarrow id$
\$E + E * E	\$	Reduce by $E \rightarrow E * E$
\$E + E	\$	Reduce $E \rightarrow E + E$
\$E	\$	accept

M.Bazargani 93-94-2

۴۲

تجزیه انتقال کاهش - پیشوند قابل وقوع

مجموعه پیشوندهای شبه جملات راست ظاهر شونده در
پشته یک تجزیه کننده انتقال کاهش

یا

پیشوندی از یک شبه جمله راست عبور نکننده از انتهای
راست سمت راست ترین دستگیره آن شبه جمله

M.Bazargani 93-94-2

۴۳

تجزیه انتقال کاهش - تداخل ها

تداخل انتقال - کاهش

تردید در عمل انتقال یا عمل کاهش در زمان تصمیم
گیری برای تجزیه کننده

تداخل کاهش کاهش

وجود چند قانون برای کاهش یک رشته در یک زمان

M.Bazargani 93-94-2

۴۴

پارسینگ تقدم عملگر Operator Precedence

گرامری را گرامر عملگر نامند اگر تنها اگر دارای شرایط زیر باشد :

- ۱-قاعده λ نداشته باشد.
- ۲-در سمت راست هیچ قاعده ای از آن گرامر دو غیرترمینال مجاور هم نباشند.

Ex:

 $E \rightarrow AB$ $A \rightarrow a$ $B \rightarrow b$

not operator grammar

 $E \rightarrow EOE$ $E \rightarrow id$ $O \rightarrow + | * | /$

not operator grammar

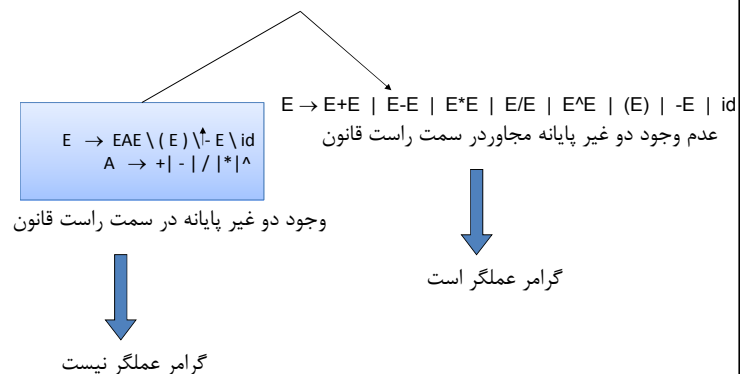
 $E \rightarrow E + E$ $E * E$ $E / E \mid id$

operator grammar

M.Bazargani ۹۳-۹۴-۲

۴۵

مثال عملگر اولویت - گرامر عملگر



M.Bazargani ۹۳-۹۴-۲

۴۶

نقطه ضعفهای روش عملگر اولویت

به دلیل محدودیتهایی که دارد عملاً گرامرهای کمی وجود دارد که بتوان از این روش برای آنها استفاده کرد. (نوع گرامر حوضه آنرا کوچک میکند.)
در مورد عملگرهایی که دو تقدم متفاوت دارند مثل عملگر " - ", " + ", " * " این روش کارایی ندارد و ممکن است برخی خطاهای نحوی را نتواند کشف کند .
روش چندان دقیقی نیست یعنی ممکن است برخی از خطاهای نحوی را نتواند کشف نماید .



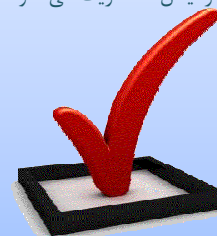
M.Bazargani 93-94-2

۴۷

پارسینگ تقدم عملگر Operator Precedence

محاسن:

در کنار نقاط ضعف بیان شده پارسینگ تقدم عملگر ویژگی بسیار مهم ساده گی پیاده سازی را دارد به همین دلیل بسیاری از کامپایلرهای موجود از این روش استفاده می کنند.
در پارسر فوق سه رابطه ی تقدم که بین ترمینال ها تعریف می شود برای هدایت عمل پارسینگ استفاده می گردد.



M.Bazargani 93-94-2

۴۸

روابط تقدم عملگر

تعريف سه رابطه اولويت مجزای بين هر زوج از پاينه ها

مفهوم	رابطه
اولويت a کمتر از b است.	$a < b$
اولويت a و b يکسان است.	$a = b$
اولويت a بيش از b است.	$a > b$

M.Bazargani 93-94-2

۴۹

نکات روابط

روابط تقدمی که در اینجا معرفی می شوند با روابط تقدمی ریاضی متفاوت هستند، بعنوان مثال در اینجا با دانستن اینکه رابطه ی $a < b$ برقرار است نمی توان نتیجه گرفت که $b > a$ می باشد.
ممکن است دو ترمینال هیچ یک از این سه نوع رابطه تقدم را با هم نداشته باشند.
یا اینکه دو ترمینال دو رابطه تقدم متفاوت داشته باشند، مثلاً در یک گرامر ممکن است هم $< *$ و هم $- < *$ باشد.



M.Bazargani 93-94-2

۵۰

نکته

از روابط اولویتی میتوان برای تشخیص دستگیره استفاده کرد
به همین منظوره ابتداوانتهای رشته w ، $\$$ را اضافه میکنیم.

$(\$w\$)$

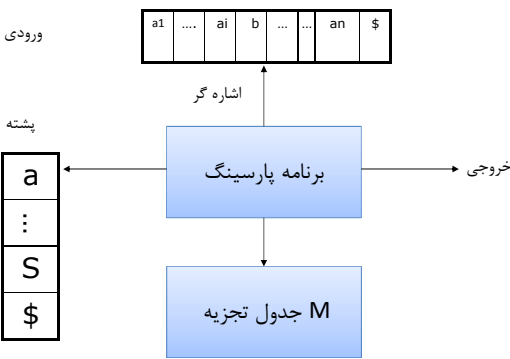
❖ اولویت $\$$ از اولویت تمام پایانی ها کمتر است.

❖ یک دستگیره اولین دنباله ایست که درپیش ازچپ به راست
رشته ورودی بین $< >$ محصور است.

M.Bazargani 93-94-2

۵۱

معماری پارسینگ تقدم عملگر



M.Bazargani ۹۳-۹۴-2

۵۲

روش کارالگوریتم پارسینگ تقدم عملگر

مولفه این پارسر یک برنامه پارسینگ است که از یک طرف ورودی خود را از اسکنر دریافت کرده و از یک پشته برای ذخیره اطلاعات و از یک جدول (جدول پارسر) برای هدایت عمل پارسینگ استفاده می نماید.

در این روش ابتدا یک علامت \$ به انتهای رشته ی ورودی اضافه می شود ، همچنین در ابتدای کار نیز پشته فقط حاوی یک علامت \$ است .

جدول پارس نیز در این روش یک جدول دو بعدی است که تعداد سطرها و ستون ها ی آن به اندازه ی تعداد ترمینال ها بعلاوه ۱ (به خاطر \$) است.

در محل هر خانه جدول حداکثر یکی از علامتهای < > = یا «هیچی» قرار می گیرد.

M.Bazargani 93-94-2

۵۳

روش کارالگوریتم پارسینگ تقدم عملگر

در هر مرحله بااستفاده از بالاترین ترمینال بالای پشته (a) و توکن جاری (b) به عنوان اندیس به جدول پارسینگ مراجعه نموده و یکی از شرایط زیر بوجود می آید:

- ۱-ممکن است $a < b$ باشد .
- ۲-ممکن است $a = b$ باشد .
- ۳-ممکن است $a > b$ باشد .
- ۴-ممکن است هیچ رابطه ای بین a و b وجود نداشته باشد .

M.Bazargani 93-94-2

۵۴

اگر $a < b$ باشد آنگاه ...

اگر $a < b$ باشد آنگاه پارسر ابتدا علامت "<" را و سپس توکن جاری b را به بالای پشته انتقال می دهد.
(عمل Shift)

M.Bazargani 93-94-2

۵۵

اگر $a = b$ باشد آنگاه ...

اگر $a = b$ باشد آنگاه پارسر فقط توکن جاری را به بالای پشته انتقال می دهد.(عمل Shift)

M.Bazargani 93-94-2

۵۶

اگر $a > b$ باشد آنگاه ...

عمل کاهش انجام می شود .
پارسر در داخل پشته آنقدر پایین می رود تا به اولین علامت "<" (خلاف علامت >) برسد.
دستگیره : رشته مابین این علامت و علامت بالای پشته است .
شرایط استثنایی هم وجود دارد : غیرپایانی زیر علامت < در صورت وجود جزء دستگیره محسوب می شود.
پارسر برای انجام عمل کاهش ؛ دستگیره پیدا شده را از پشته حذف نموده و به جای آن غیرترمینال سمت چپ قاعده مورد استفاده قرار می گیرد.

M.Bazargani 93-94-2

۵۷

روش کار الگوریتم پارسینگ تقدم عملگر

❖ در صورتیکه در پشته زیرعلامت فوق (<) غیرپایانی موجود باشد مثل A، آن را هم به دستگیره اضافه می کند. در این مرحله پارسر برای انجام عمل کاهش دستگیره ی پیداشده را از پشته حذف نموده و به جای آن غیرترمینال سمت چپ قاعده مورد استفاده قرار می گیرد.
❖ اگر ترمینال بالای پشته با ورودی جاری رابطه ای نداشته باشد ، یعنی سطر a و ستون bام در جدول پارس خالی باشد ، آنگاه یک خطای نحوی رخ داده و پارسر رویه ی اصلاح خطا را فراخوانی می کند.

M.Bazargani 93-94-2

۵۸

الگوریتم پارسینگ تقدم عملگر

The input string is w\$, the initial stack is \$ and a table holds precedence relations between certain terminals

Algorithm:

```
set p to point to the first symbol of w$ ;
repeat forever
  if ( $ is on top of the stack and p points to $ ) then return
  else {
    let a be the topmost terminal symbol on the stack and let b be the
    symbol pointed to by p;
    if (  $a < b$  or  $a = b$  ) then { /* SHIFT */
      push b onto the stack;
      advance p to the next input symbol;
    }
    else if (  $a > b$  ) then /* REDUCE */
      repeat pop stack
      until ( the top of stack terminal is related by  $<$  to the terminal most
      recently popped );
    else error();
  }
} M.Bazargani 93-94-2
```

۵۹

الگوریتم پارسینگ تقدم عملگر

```
(1) set ip to point to the first symbol of w$;
(2) repeat forever
(3)   if $ is on top of the stack and ip points to $ then
(4)     return
(5)   else begin
(6)     let a be the topmost terminal symbol on the stack
(7)     and let b be the symbol pointed to by ip;
(8)     if  $a < b$  or  $a = b$  then begin
(9)       push b onto the stack;
(10)      advance ip to the next input symbol;
(11)    end;
(12)    else if  $a > b$  then /* reduce */
(13)      repeat
(14)        pop the stack
(15)      until the top stack terminal is related by  $<$ 
(16)        to the terminal most recently popped
(17)    else error()
(18)  end
```

مثالی از یک جدول پارسینگ

- گرامر مقابل را در نظر بگیرید :
$$E \rightarrow E+E \mid E-E \mid E^*E \mid E/E \mid E^{\wedge}E \mid (E) \mid -E \mid id$$
- فرض کنید رشته ورودی $id+id*id$ با تقدم اولویت زیر مشخص شده است :
$$\$ < id > + < id > * < id > \$$$
- جدول پارسینگ تقدم عملگر به شکل زیر خواهد بود :

	id	+	*	\$
id	e	>	>	>
+	<	>	<	>
*	<	>	>	>
\$	<	<	<	e

M.Bazargani 93-94-2

۶۱

برای پیدا کردن دستگیره

رشته ورودی را از چپ تا زمانیکه به اولین $>$ برسد اسکن می کنیم .
سپس به پویش معکوس به چپ تا جاییکه به $=$ و $<$ برسد می پردازیم .
دستگیره شامل هر چیزی در سمت چپ $>$ و سمت راست $<$ می باشد

$\$ < id > + < id > * < id > \$$	$E \rightarrow id$	$\$ E + id * id \$$
$\$ E + < id > * < id > \$$	$E \rightarrow id$	$\$ E + E * id \$$
$\$ E + E * < id > \$$	$E \rightarrow E+E$	$\$ E + E * E \$$
$\$ < E * > id > \$$	$E \rightarrow E$	$\$ E * id \$$
$\$ < E * E > \$$	$E \rightarrow E^*E$	$\$ E \$$
$\$ \$$		$\$ \$$

M.Bazargani 93-94-2

۶۲

پیدا کردن روابط تقدم عملگرها و تشکیل جدول پارسینگ

جدول پارسینگ تقدم عملگر از پیدا کردن روابط تقدم میان عملگرها بوجود می آید .
برای تعیین روابط تقدم عملگر از دو تابع استفاده می کنیم :

$$\text{Firstterm}(A) = \{c \mid A \Rightarrow c\alpha \text{ OR } A \Rightarrow Bc\alpha\}$$

$$\text{Lastterm}(A) = \{a \mid A \Rightarrow \alpha a \text{ or } A \Rightarrow \alpha a B\}$$

c یک پایانی و B یک غیرترمینال و α یک رشته از پایانی ها و غیرپایانی ها می باشد .

M.Bazargani 93-94-2

۶۳

Firstterm(A) & Lastterm(A)

❖ **Firstterm(A)**: مجموعه ای از پایانی ها است که در سمت چپ ترین بخش تولیدات A قرار دارند یا به فاصله یک غیر پایانی در سمت چپ ترین بخش تولیدات A قرار گرفته اند.

❖ **Lastterm(A)**: مجموعه ای از پایانی ها است که در سمت راست ترین بخش تولیدات A قرار دارند یا به فاصله یک غیر پایانی در سمت راست ترین بخش تولیدات A قرار می گیرند .

M.Bazargani 93-94-2

۶۴

نکات برای تعیین Firstterm و Lastterm

اگر قاعده ای به شکل $A \rightarrow BaC$ داشته باشیم آنگاه اعضای $Firstterm(B)$ در $Firstterm(A)$ قرار دارند.
اگر قاعده ای به شکل داشته باشیم آنگاه اعضای $Lastterm(C)$ در $Lastterm(A)$ قرار دارند.

M.Bazargani 93-94-2

۱۰

مثال

مجموعه $Firstterm$ و $Lastterm$ غیرپایانی های زیر را بدست آورید.

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

	E	T	F
Firstterm	+, *, (, id	*, (, id	(, id
Lastterm	+, *,), id	*,), id	), id

M.Bazargani ۹۳-۹۴-۲

۱۱

قواعد تکمیل جدول پارسینگ

پس از پیدا نمودن توابع $Firstterm$ و $Lastterm$ برای هر غیر پایانی با استفاده از سه قاعده زیر می توان روابط بین پایانی ها را پیدا نمود:

- قاعده ۱: $IF \exists U \rightarrow \dots aWb \dots \text{ Or } U \rightarrow \dots ab \dots \text{ Then } a = b$
 قاعده ۲: $IF \exists U \rightarrow \dots aW \dots \text{ and } b \in Firstterm(W) \text{ Then } a < b$
 قاعده ۳: $IF \exists U \rightarrow \dots Wb \dots \text{ and } a \in Lastterm(W) \text{ Then } a > b$

M.Bazargani 93-94-2

۱۲

ادامه حل مثال از پارسینگ تقدم عملگر

❖ با استفاده از $Firstterm$ و $Lastterm$ که در قسمت قبلی بدست آمد روابط میان ترمینالها را بدست آورید.

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

❖ برای تشکیل جدول کافی است اگر رابطه ای به شکل aRb داریم به طوری که a و b ترمینال بوده و R عملگر مقایسه ای است در ردیف a و ستون b جدول پارسینگ عملگر R را قرار دهیم.

M.Bazargani ۹۳-۹۴-۲

۱۳

جدول پارسینگ

	+	*	(	)	id	\$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	error
)	>	>	error	>	error	>
id	>	>	error	>	error	>
\$	<	<	<	error	<	=

M.Bazargani ۹۳-۹۴-۲

۶۹

اولویت های بدیهی

- ۱- اگر عملگر A اولویت بیشتر از عملگر B داشته باشد، روابط $A > B$ و $A < B$ بین آنها برقرار است.
- ۲- اگر A و B عملگرهای با اولویت یکسان باشند، اگر هر دو شرکت پذیر از راست هستند: $B < A$, $A < B$ و شرکت پذیر از چپ: $B > A$, $A > B$

M.Bazargani ۹۳-۹۴-۲

۷۰

اولویت های بدیهی

- ۳- برای تمام عملگرهای مانند A روابط زیر برقرار است.

$\$ < A$ $A > \$$ $A >)$ $) > A$ $(< A$
 $A < ($ $id > A$ $A < id$

- ۴- روابط زیر همیشه برقرارند:

$(=)$ $(< ($ $(< id$
 $\$ < ($ $id > \$$ $id >)$
 $\$ < id$ $) > \$$ $) >)$

M.Bazargani ۹۳-۹۴-۲

۷۱

پارسر تقدم ساده Simple Precedence Parser

شبیه به پارسر تقدم عملگر است. با این تفاوت که در این روش روابط بین تمام عناصر تعریف شده بررسی می شود ولی در روش تقدم عملگر فقط بین پایانه ها تعریف می شود. توسعه یافته تقدم عملگر است. محدودیت کمتری نسبت به گرامرهای تقدم عملگر دارد که باعث می شود طیف بیشتری از گرامرها را در برگیرد. شرایط گرامرهای تقدم ساده:

- ۱- قاعده ϵ نداشته باشد. (حذف قواعد ϵ و حذف قواعد پوچ)
 ۲- سمت راست هیچ دو قاعده ای از گرامر نباید یکسان باشد.

M.Bazargani ۹۳-۹۴-۲

۷۲

پارسر تقدم ساده (ادامه ...)

در این پارسر هم نیاز به تعریف رابطه تقدم میان پایانی ها و غیرپایانی ها است که از علائم < و = و > استفاده می شود.

جدول پارس یک تقدم ساده یک جدول به تعداد تعداد پایانی ها و غیرپایانی ها به علاوه یک سطر و ستون است. $(T + V + 1)$

برای تعیین روابط تقدم ساده از دو تابع به نامهای Head و Tail استفاده می شود.

M.Bazargani ۹۳-۹۴-۲

۷۳

پارسر تقدم ساده (ادامه ...)

گرچه گرامرهای تقدم ساده ؛ دسته وسیع تری از گرامرها را نسبت به گرامرهای تقدم عملگر می پوشانند اما نمی توان گفت که گرامرهای تقدم عملگر زیرمجموعه گرامرهای تقدم ساده هستند و برعکس .

الف $S \rightarrow (SS)|c$

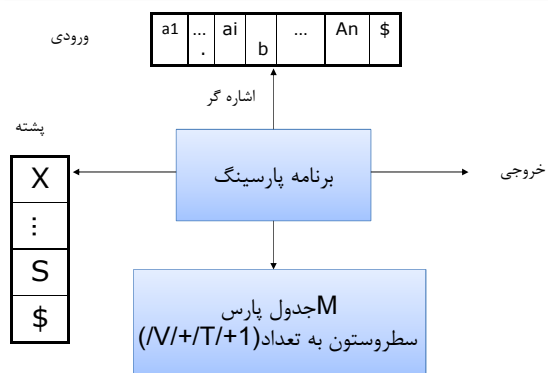
ب $S \rightarrow (A) \quad A \rightarrow (A)A|(A)|id$

گرامر الف تقدم ساده است و اما تقدم عملگر نیست و همچنین گرامر ب تقدم عملگر است اما تقدم ساده نیست؛ چون دو غیر پایانی در کنار هم قرار گرفته اند و گرامر ب نیز تقدم ساده نیست چون سمت راست دو قاعده با هم یکی است .

M.Bazargani ۹۳-۹۴-۲

۷۴

معماری پارسینگ تقدم ساده



M.Bazargani ۹۳-۹۴-۲

۷۵

تابع Head(A)

ورودی: غیرترمینال گرامر تقدم ساده
خروجی: مجموعه ای از غیرترمینالها و ترمینالهای گرامر
تعریف: عبارتست از مجموعه ای از ترمینالها و غیرترمینالها که درست چپ ترین محل عبارات تولید شده در اشتقاق بعدی که قواعدی که A در سمت چپ آنها قرار دارد وجود دارد .

$$\text{Head}(A) = \{ Y | A \Rightarrow Y\alpha, Y \in (V \cup T)^* \}$$

$$\{ A \Rightarrow Y\alpha ; A \in V ; \alpha \in (V \cup T)^* \}$$

M.Bazargani ۹۳-۹۴-۲

۷۶

تابع Tail(A)

ورودی: غیرترمینال گرامر تقدم ساده
خروجی: مجموعه ای از غیرترمینالها و ترمینالهای گرامر
تعریف: عبارتست از مجموعه ترمینالها و غیرترمینالهایی که در سمت
راست ترین محل عبارات تولیدشده در اشتقاق قواعدی که A در سمت
چپ آنها قراردارند وجود دارد.

$$\text{Tail}(A) = \{ Y | A \Rightarrow \alpha Y \}$$

$$\text{VUT}^*; Y \in (\text{VUT}) \in \alpha \{$$

M.Bazargani ۹۳-۹۴-۲

۷۷

مثال

جدول پارس عملگر ساده گرامر مقابل را بدست آورید.

$$M \rightarrow \$SS \quad S \rightarrow (SS)|c$$

	s	m
Head	(,c	\$
Tail	),c	\$

M.Bazargani ۹۳-۹۴-۲

۷۸

پارسر تقدم ساده (نکات) ...

برای بدست آوردن مجموعه Head و Tail یک گرامر جهت
شامل شدن علامت \$ در جدول پارس قاعده زیر را به ابتدای
گرامر می افزاییم به طوری که S عنصر شروع گرامر باشد.

$$M \rightarrow \$SS$$

اگر قاعده ای به شکل $A \rightarrow B\alpha C$ داشته باشیم آنگاه اعضای
Head(B) در Head(A) و اعضای Tail(C) در Tail(A)
قرار دارد.

M.Bazargani ۹۳-۹۴-۲

۷۹

محاسبه روابط بین پایانی ها و غیرپایانی ها

$$1) X = Y \text{ iff } \exists u \rightarrow \dots XY \dots$$

رابطه X و Y رابطه = می باشد اگر و تنها اگر در یک فرم جمله ای در کنار
هم ظاهر شوند. (X و Y می توانند پایانی و غیرپایانی باشند)

$$X, Y \in (\text{VUT})$$

$$2) X < Y \text{ iff } \exists u \rightarrow \dots XA \dots \text{ and } Y \in \text{Head}(A)$$

رابطه X و Y رابطه < می باشد اگر و تنها اگر در یک فرم جمله ای
X قبل از یک پایانه ظاهر شود و Y عضو مجموعه Head آن
غیر پایانه است.

$$X(\text{VUT}); A \in V$$

M.Bazargani ۹۳-۹۴-۲

۸۰

محاسبه روابط بین پایانی ها و غیرپایانی ها

$$3) X > Y \text{ iff } \exists u \rightarrow \dots AB \dots \text{ and } X \in \text{Tail}(A) \text{ and } (Y \in \text{Head}(B) \text{ or } Y = B)$$

رابطه بین X و Y رابطه $>$ برقرار است اگر در یک فرم جمله ای دو غیرپایانه AB در کنار هم قرار گرفته باشند و X متعلق به $\text{Tail}(A)$ باشد و یا Y با B برابر و یا متعلق به $\text{Head}(B)$ باشد.

$$\{U \rightarrow \dots AB \dots; \text{Tail}(A) > \text{Head}(B)\}$$
$$\{A, B \in V\}$$

مثال

جدول پارس عملگر ساده گرامر مقابل را بدست آورید .

$$M \rightarrow \$SS \quad S \rightarrow (SS) | c$$

	\$	(	)	c	\$
\$	=	<	=	<	=
(	=	<	error	<	error
)	>	>	>	>	>
c	>	>	>	>	>
\$	=	<	error	<	error

	S	M
Head	(,c	\$
Tail	),c	\$

قوانین جدول پارس عملگر ساده

$X=Y$:if
1) $u \rightarrow \dots XY \dots$
 $X, Y \in (VUT)$

$X<Y$:if
2) $U \rightarrow \dots XA \dots$
 $X \in (VUT)$
 $Y \in \text{head}(A)$
 $A \in V$

$X>Y$
3) $U \rightarrow \dots AB \dots$
 $A, B \in V$
 $X \in \text{tail}(A)$
 $Y \in \text{head}(B)$

OR

$X>Y$
3) $U \rightarrow \dots AY \dots$
 $A \in V$
 $X \in \text{tail}(A)$
 $Y \in (VUT)$

مثال قوانین جدول پارس عملگر ساده

○ برای قانون اول داریم:

$S=($ $S=)$ $\$=S$ $S=\$$

مثال قوانین جدول پارس عملگر ساده

برای قانون دوم داریم :

$$\$ < \text{head}(s) \Rightarrow \$ < \{ (, c \}$$

$$(< \text{head}(s) \Rightarrow (< \{ c, c \}$$

$$S < \text{head}(s) \Rightarrow S < \{ (, c \}$$

M.Bazargani ۹۳-۹۴-۲

۸۵

مثال قوانین جدول پارس عملگر ساده

برای قانون سوم داریم:

$$\text{Tail}(s) > \$ \Rightarrow \{), c \} > \$$$

$$\text{Tail}(s) >) \Rightarrow \{), c \} >)$$

M.Bazargani 93-94-2

۸۶

مراحل اجرای الگوریتم تقدم ساده

❖ توکن جاری b و عنصر بالای پشته X (که می تواند پایانه یا غیر پایانی باشد) می باشد .

۱- در صورتیکه $X < b$ باشد ابتدا $<$ و سپس b به بالای پشته انتقال می یابد.

۲- در صورتیکه $X = b$ باشد پارسر فقط b را به بالای پشته انتقال می دهد .
❖ در صورتیکه رابطه ای با هم نداشته باشند یک خطای نحوی رخ می دهد.

❖ در صورتیکه $b = \$$ و در داخل پشته $\$S$ باقی مانده باشد تجزیه با موفقیت پایان یافته است .

۸۷

مراحل اجرای الگوریتم تقدم ساده

❖ در صورتیکه $X > b$ باشد پارسر عمل کاهش را انجام می دهد. و در این حالت دستگیره برشته بالای پشته تا اولین علامت $<$ است .

ابتدا دستگیره را از بالای پشته حذف می کند .

عنصر بالای پشته را پس از حذف دستگیره Top می نامیم و سمت چپ تاوردی را که پارسر از آن جهت کاهش استفاده می کند Lhs می نامیم .
روابط Top و Lhs را با هم بررسی می کنیم .

M.Bazargani ۹۳-۹۴-۲

۸۸

بررسی روابط Top و Lhs

پارسر Top و Lhs را از جدول استخراج نموده و یکی از اعمال زیر را انجام می دهد :

۱- اگر $Top < Lhs$ باشد پارسر ابتدا علامت $<$ و سپس یک توکن از ورودی را وارد پشته می کند .

۲- اگر $Top = Lhs$ باشد پارسر فقط یک توکن از ورودی را وارد پشته می کند.

۳- اگر Top و Lhs رابطه ای نداشته باشند یک خطای نحوی رخ داده است و باید رویه خطا را اصلاح کند.

M.Bazargani 93-94-2

۸۹

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))$	Shift

M.Bazargani

۹۰

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc)\$$	Shift

M.Bazargani 93-94-2

۹۱

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc)\$$	Shift
2	$\$ < (< c$	$(cc)\$$	Reduce $S \rightarrow c$

M.Bazargani

۹۲

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc)\$$	Shift
2	$\$ < (< c$	$(cc)\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc)\$$	shift

M.Bazargan

۹۳

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc)\$$	Shift
2	$\$ < (< c$	$(cc)\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc)\$$	Shift
4	$\$ < (S < ($	$cc)\$$	Shift

M.Bazargan

۹۴

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc)\$$	Shift
2	$\$ < (< c$	$(cc)\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc)\$$	Shift
4	$\$ < (S < ($	$cc)\$$	Shift
5	$\$ < (S < (< c$	$c)\$$	Reduce $S \rightarrow c$

M.Bazargan

۹۵

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc)\$$	Shift
2	$\$ < (< c$	$(cc)\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc)\$$	Shift
4	$\$ < (S < ($	$cc)\$$	Shift
5	$\$ < (S < (< c$	$c)\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c)\$$	Shift

M.Bazargan

۹۶

تجزیه جمله $c(cc)$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc))\$$	Shift
2	$\$ < (< c$	$(cc))\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc))\$$	Shift
4	$\$ < (S < ($	$cc))\$$	Shift
5	$\$ < (S < (< c$	$c))\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c))\$$	Shift
7	$\$ < (S < (S < c$	$)\$$	Reduce $S \rightarrow c$

M.Bazargan

۹۷

تجزیه جمله $c(cc)$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc))\$$	Shift
2	$\$ < (< c$	$(cc))\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc))\$$	Shift
4	$\$ < (S < ($	$cc))\$$	Shift
5	$\$ < (S < (< c$	$c))\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c))\$$	Shift
7	$\$ < (S < (S < c$	$)\$$	Reduce $S \rightarrow c$
8	$\$ < (S < (SS$	$)\$$	Shift

M.Bazargan

۹۸

تجزیه جمله $c(cc)$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc))\$$	Shift
2	$\$ < (< c$	$(cc))\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc))\$$	Shift
4	$\$ < (S < ($	$cc))\$$	Shift
5	$\$ < (S < (< c$	$c))\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c))\$$	Shift
7	$\$ < (S < (S < c$	$)\$$	Reduce $S \rightarrow c$
8	$\$ < (S < (SS$	$)\$$	Shift
9	$\$ < (S < (SS)$	$)\$$	Reduce $S \rightarrow (SS)$

M.Bazargan

۹۹

تجزیه جمله $c(cc)$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc))\$$	Shift
2	$\$ < (< c$	$(cc))\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(cc))\$$	Shift
4	$\$ < (S < ($	$cc))\$$	Shift
5	$\$ < (S < (< c$	$c))\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c))\$$	Shift
7	$\$ < (S < (S < c$	$)\$$	Reduce $S \rightarrow c$
8	$\$ < (S < (SS$	$)\$$	Shift
9	$\$ < (S < (SS)$	$)\$$	Reduce $S \rightarrow (SS)$
10	$\$ < (SS$	$)\$$	shift

M.Bazargan

۱۰۰

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc))\$$	Shift
2	$\$ < (< c$	$(cc))\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(c c))\$$	Shift
4	$\$ < (S < ($	$cc))\$$	Shift
5	$\$ < (S < (< c$	$c))\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c))\$$	Shift
7	$\$ < (S < (S < c$	$))\$$	Reduce $S \rightarrow c$
8	$\$ < (S < (SS$	$))\$$	Shift
9	$\$ < (S < (SS)$	$) \$$	Reduce $S \rightarrow (SS)$
10	$\$ < (SS$	$) \$$	Shift
11	$\$ < (SS)$	$\$$	Reduce $S \rightarrow (SS)$

۱۰۱

تجزیه جمله $(c(cc))$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انجام شده
0	\$	$(c(cc))\$$	Shift
1	$\$ < ($	$c(cc))\$$	Shift
2	$\$ < (< c$	$(cc))\$$	Reduce $S \rightarrow c$
3	$\$ < (S$	$(c c))\$$	Shift
4	$\$ < (S < ($	$cc))\$$	Shift
5	$\$ < (S < (< c$	$c))\$$	Reduce $S \rightarrow c$
6	$\$ < (S < (S$	$c))\$$	Shift
7	$\$ < (S < (S < c$	$))\$$	Reduce $S \rightarrow c$
8	$\$ < (S < (SS$	$))\$$	Shift
9	$\$ < (S < (SS)$	$) \$$	Reduce $S \rightarrow (SS)$
10	$\$ < (SS$	$) \$$	Shift
11	$\$ < (SS)$	$\$$	Reduce $S \rightarrow (SS)$
12	$\$ < S$		Accept

M.Bazargan

۱۰۲

مثال

• گرامر مقابل را در نظر بگیرید :

$S \rightarrow +AB \mid -$ $A \rightarrow (A * | *$ $B \rightarrow c$

	S	A	B
Head	$+, -$	$(, *$	c
Tail	$-, c, B$	*	c

M.Bazargani 93-94-2

۱۰۳

جدول پارس

	S	A	B	\$	+	-	(	*	c
S				=					
A			=					=	<
B				>					
\$	=				<	<			
+		=					<	<	
-				>					
(		=					<	<	
*								>	>
c				>					

۱۰۴

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +

M.B

۱۰۵

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	\$ < +	$(C^{**}$	Shift (

M.Bazargani 93-94-2

۱۰۶

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	\$ < +	$(C^{**}$	Shift (
2	\$ < + < (	$^{**}C$	Shift *

M.Bazargani 90-91-2

۱۰۷

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	\$ < +	$(C^{**}$	Shift (
2	\$ < + < (	$^{**}C$	Shift *
3	\$ < + < (< *	C^{**}	Reduce $A \rightarrow *$

M.Bazargani 93-94-2

۱۰۸

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	$\$ < +$	$(C^{**}$	Shift (
2	$\$ < + < ($	C^{**}	Shift *
3	$\$ < + < (< *$	C^*	Reduce $A \rightarrow *$
4	$\$ < + < (A$	C^*	Shift *

M.Bazargani 93-94-2

۱۰۹

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	$\$ < +$	$(C^{**}$	Shift (
2	$\$ < + < ($	C^{**}	Shift *
3	$\$ < + < (< *$	C^*	Reduce $A \rightarrow *$
4	$\$ < + < (A$	C^*	Shift *
5	$\$ < + < (A *$	C	Reduce $A \rightarrow (A*$

M.B

۱۱۰

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	$\$ < +$	$(C^{**}$	Shift (
2	$\$ < + < ($	C^{**}	Shift *
3	$\$ < + < (< *$	C^*	Reduce $A \rightarrow *$
4	$\$ < + < (A$	C^*	Shift *
5	$\$ < + < (A *$	C	Reduce $A \rightarrow (A*$
6	$\$ < + A$	C	Shift c

M.Bazargani 93-94-2

۱۱۱

تجزیه جمله $C^{**}+$ برای گرامر قبلی ...

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+C^{**}$	Shift +
1	$\$ < +$	$(C^{**}$	Shift (
2	$\$ < + < ($	C^{**}	Shift *
3	$\$ < + < (< *$	C^*	Reduce $A \rightarrow *$
4	$\$ < + < (A$	C^*	Shift *
5	$\$ < + < (A *$	C	Reduce $A \rightarrow (A*$
6	$\$ < + A$	C	Shift c
7	$\$ < + A < c$	$\$$	Reduce $B \rightarrow c$

M.Bazargani 93-94-2

۱۱۲

برای گرامر قبلی $C...^{**}+$ تجزیه جمله

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+(**c\$$	Shift +
1	$\$ < +$	$(**c\$$	Shift (
2	$\$ < + < ($	$**c\$$	Shift *
3	$\$ < + < (< *$	$*c\$$	Reduce $A \rightarrow *$
4	$\$ < + < (A$	$*c\$$	Shift *
5	$\$ < + < (A *$	$c\$$	Reduce $A \rightarrow A*$
6	$\$ < + A$	$c\$$	Shift c
7	$\$ < + A < c$	$\$$	Reduce $B \rightarrow c$
8	$\$ < + AB$	$\$$	Reduce $S \rightarrow +AB$

M.E

۱۳

برای گرامر قبلی $C...^{**}+$ تجزیه جمله

مرحله	وضعیت پشته	ورودی	عمل انتخاب شده
0	\$	$+(**c\$$	Shift +
1	$\$ < +$	$(**c\$$	Shift (
2	$\$ < + < ($	$**c\$$	Shift *
3	$\$ < + < (< *$	$*c\$$	Reduce $A \rightarrow *$
4	$\$ < + < (A$	$*c\$$	Shift *
5	$\$ < + < (A *$	$c\$$	Reduce $A \rightarrow A*$
6	$\$ < + A$	$c\$$	Shift c
7	$\$ < + A < c$	$\$$	Reduce $B \rightarrow c$
8	$\$ < + AB$	$\$$	Reduce $S \rightarrow +AB$
9	$\$ \$$	$\$$	Accept

M.Bazargani 93-94-2

۱۱۴

تجزیه گر LR

❖ تجزیه گر $LR(1)$ حالت خاصی از تجزیه گر $LR(k)$ است.

❖ L : پویش رشته ورودی از چپ به راست.

❖ R : عکس اشتقاق راست.

❖ K : توکن هایی است که برای انتخاب عمل صحیح استفاده می شود.

M.Bazargani 93-94-2

۱۱۵

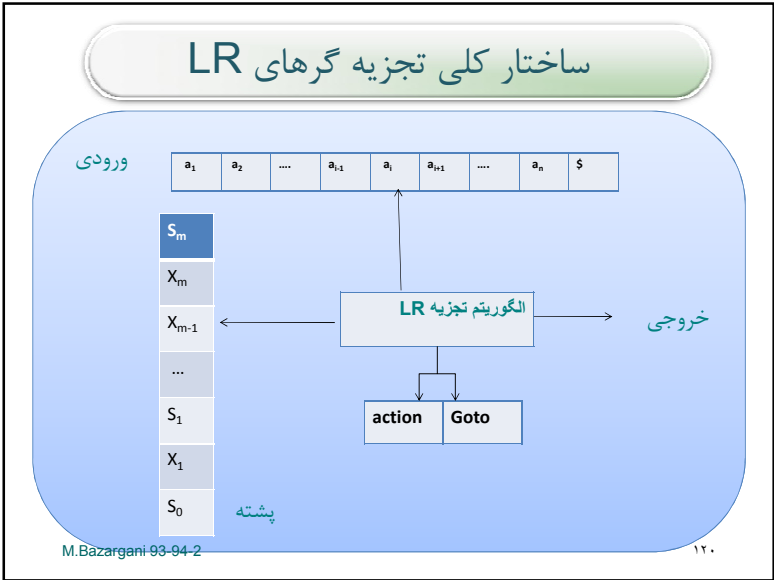
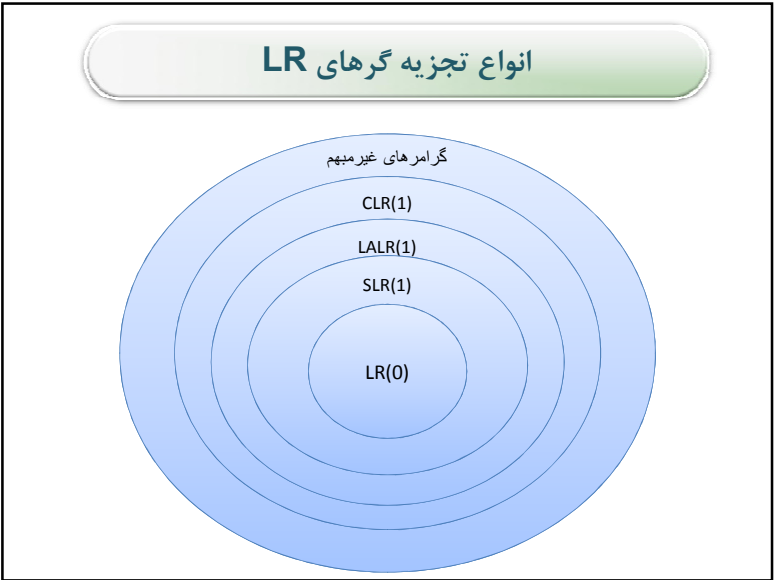
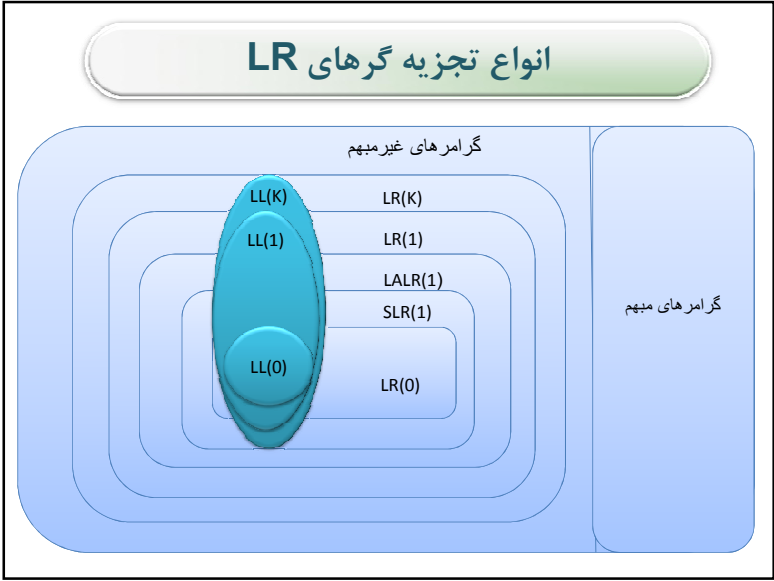
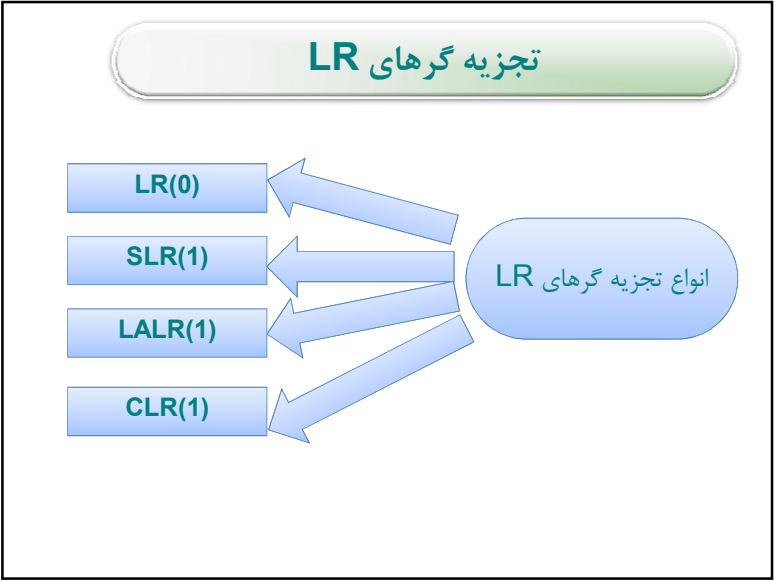
تجزیه گر LR

❖ مزایای این تجزیه گرها این است که نسبت به تجزیه گرهای پیشگوی پایین به بالا قدرت بیشتری دارند و گرامرهای بیشتری را پویش می کنند و همچنین از نظر تشخیص خطا دقت بالاتری دارند بطوری که خطای نحوی را در محل وقوع خطا تشخیص می دهند.

❖ عیب این تجزیه گرها این است که پیاده سازی دستی آنها مشکل است

M.Bazargani 93-94-2

۱۱۶



قسمتهای مختلف معماری پارسر LR

ورودی:

رشته ورودی که پارسر باید آن را تجزیه کند، در این قسمت قرار دارد.

❖ پارسر LR با دیدن \$ در ورودی پایان تجزیه را اعلام میکند.

پشته:

محتوای پشته به صورت $S.X_1S_1 \dots X_{m-1}S_{m-1}X_mS_m$ است. S_1 ها نشان دهنده ی حالات X_j ها نشان دهنده ی پایانی ها و غیر پایانی هاست.

M.Bazargani 93-94-2

۱۲۱

قسمتهای مختلف معماری پارسر LR

❑ توان های ورودی به پشته ، انتقال (shift) داده میشود تا یک دستگیره تشخیص داده شود.

❑ بعد از تشخیص دستگیره عمل کاهش (Reduce) صورت میگیرد.

- جدول تجزیه شامل دوبخش action و goto است. action برای انتخاب عمل صحیح استفاده میشود. goto حالت بعدی را مشخص میکند.

ساختار کلی تجزیه گرهای LR

❖ بخش action به تعداد حالات سطر و به تعداد پایانه ها به علاوه یک ستون دارد و بخش goto به تعداد حالات سطر و به تعداد غیر پایانه ها ستون دارد.

❖ اندازه جدول تجزیه:

$$|PT| = |action| + |goto| = (|T| + |N| + 1) * \text{تعداد حالات}$$

M.Bazargani 93-94-2

۱۲۲

ساختار کلی تجزیه گرهای LR

محتوای هر خانه action حاوی دستور انتقال و به دنبال آن یک شماره حالت، دستور کاهش و به دنبال آن یک شماره قاعده یا خالی بوده و یک خانه از بخش action حاوی کلمه accept است. هر خانه goto یا خالی بوده یا حاوی یک شماره حالت است.

M.Bazargani 93-94-2

۱۲۳

پیکربندی

پیکربندی وضعیت تجربه گر را نشان میدهد و یک زوج مرتب است که مولفه اول آن محتویات پشته بوده و مولفه دوم آن بخش باقیمانده ورودی است:

$$\langle S, X_1 S_1 X_2 S_2 \dots X_{m-1} S_{m-1} X_m S_m, a_i a_{i+1} \dots a_n \$ \rangle$$



M.Bazargani ۹۳-۹۴-2

۱۲۵

پیکربندی

اگر در مولفه اول S_i ها برای $i=0 \dots m$ را حذف کنیم و مولفه دوم یعنی بخش مصرف نشده ورودی را به آن بچسبانیم فرم جمله فعلی بدست می آید که از $w\$$ در طی صفر یا چند قدم کاهش به این فرم جمله رسیدیم:

$$X_1 X_2 \dots X_{m-1} X_m a_i a_{i+1} \dots a_n \$ \rightarrow w \$$$

و با چند قدم کاهش از این فرم جمله فعلی به SS خواهیم رسید.

$$SS \rightarrow X_1 X_2 \dots X_{m-1} X_m a_i a_{i+1} \dots a_n \$$$

M.Bazargani ۹۳-۹۴-2

۱۲۶

انواع تجزیه گرهای LR

❖ در تجزیه گر $LR(0)$ از صفر توکن برای انتخاب عمل صحیح استفاده می شود. در سایر تجزیه گرهای فوق از یک توکن برای انتخاب عمل صحیح استفاده می شود.

❖ الگوریتم تجزیه برای همه این تجزیه گرها یکسان است و تفاوتشان در نحوه ساخت جدول تجزیه و محتویات خانه های جدول تجزیه است.

M.Bazargani ۹۳-۹۴-2

۱۲۷

تجزیه گر LR

برنامه تجزیه گر LR با استفاده از محتویات جدول تجزیه گر، عمل مناسب را انتخاب میکند.

: shift (۱)
Action[i,a]=Sn

ابتدا n و سپس n رابه پشته انتقال میدهد.

:Reduce (۲)
Action[i,a]=rn
یک دستگیره یافت شده که باید کاهش یابد.

M.Bazargani ۹۳-۹۴-2

۱۲۸

تجزیه گره LR

✓ اگر n شماره قاعده تولید $A \rightarrow \beta$ باشد برنامه تجزیه کننده دنباله β را از بالای پشته حذف میکند.

پس از حذف β اگر حالت بالای پشته m باشد برنامه تجزیه کننده ابتدا A و سپس عدم وجود در $\text{goto}[m, A]$ را به بالای پشته اضافه میکند.

نکته: برای حذف β از بالای پشته تعداد $2/\beta$ از بالای پشته حذف میگردد. زیرا بین هر نماد در پشته یک شماره حالت نیز وجود دارد.

M.Bazargani ۹۳-۹۴-۲

۱۲۹

دسته بندی زبانها



M.Bazargani ۹۳-۹۴-۲

۱۳۰

انواع تجزیه گرهای LR

با توجه به شکل فوق نکات زیر نتیجه گیری میشود:

- ❖ هر گرامر $LR(0)$ یک گرامر $SLR(1)$ است.
- ❖ هر گرامر $SLR(1)$ یک گرامر $LALR(1)$ است.
- ❖ هر گرامر $LALR(1)$ یک گرامر $CLR(1)$ است.
- ❖ هر گرامر $SLR(1)$ یک گرامر $CLR(1)$ است.
- ❖ هر گرامر $CLR(1)$ یک گرامر غیر مبهم است.

M.Bazargani ۹۳-۹۴-۲

۱۳۱

الگوریتم تجزیه $LR(1)$

❖ با فرض اینکه در انتهای رشته ورودی علامت $\$$ قرار گرفته است تجزیه گر در شروع کار حالت صفر در پشته خالی قرار می دهد و اسکنر را برای دریافت اولین توکن فراخوانی می کند. اگر حالت بالای پشته را S_m و توکن جاری را a_i بنامیم، تجزیه گر در طی یک حلقه با مراجعه به بخش action تقاطع سطر و ستون عمل صحیح را انتخاب می کند و یکی از اعمال زیر را انجام می دهد:

M.Bazargani ۹۳-۹۴-۲

۱۳۲

الگوریتم تجزیه LR(1)

الف) اگر $\text{action}[S_m, a_i]$ برابر shift S باشد آنگاه عمل انتقال انجام میشود بدین صورت که تجزیه گر را در بالای پشته قرار داده و روی آن حالت S را قرار می دهد. پیکربندی قبل و بعد از عمل انتقال به صورت زیر است:

پیکره بندی فعلی:

$$< S_1 X_1 S_1 X_1 S_1 \dots X_m S_m a_i a_{i+1} \dots a_n \$ >$$

پیکربندی جدید:

$$< S_1 X_1 S_1 X_1 S_1 \dots X_m S_m a_i S_{i+1} a_{i+1} \dots a_n \$ >$$

M.Bazargani ۹۳-۹۴-۲

۱۳۳

الگوریتم تجزیه LR(1)

ب) اگر $\text{action}[S_m, a_i]$ برابر reduce $A \rightarrow \alpha$ باشد آنگاه عمل کاهش انجام میشود. برای اینکه دستگیره α از بالای پشته حذف شود باید به اندازه طول دستگیره α علامت X از بالای پشته حذف شود. اما در بالای پشته یک در میان حالت و علامت است پس به اندازه دو برابر طول دستگیره باید از بالای پشته نماد برداشته شود. اگر $|A| = r$ بنامیم:

پیکربندی جدید:

$$< S_1 X_1 S_1 X_1 S_1 \dots X_{m-r} S_{m-r} A S_{i+1} a_{i+1} \dots a_n \$ >$$

در حالی که $\text{goto}[S_{m-r}, A]$ برابر S است.

M.Bazargani ۹۳-۹۴-۲

۱۳۴

الگوریتم تجزیه LR(1)

وقتی به اندازه دو برابر r نماد از بالای پشته حذف شود:

$$S_1 X_1 S_1 X_1 S_1 \dots X_{m-r} S_{m-r}$$

بعد از عمل کاهش پشته بصورت زیر در میاید:

$$S_1 X_1 S_1 X_1 S_1 \dots X_{m-r} S_{m-r} A$$

اگر فرض شود $\text{goto}[S_{m-r}, A]$ برابر S باشد حالت S روی A بالای پشته قرار داده می شود.

پیکربندی جدید:

$$< S_1 X_1 S_1 X_1 S_1 \dots X_{m-r} S_{m-r} A S_{i+1} a_{i+1} \dots a_n \$ >$$

M.Bazargani ۹۳-۹۴-۲

۱۳۵

الگوریتم تجزیه LR(1)

ج) اگر $\text{action}[S_m, a_i]$ برابر accept باشد، آنگاه تجزیه گریا موفقیت به کارش خاتمه میدهد.

د) اگر $\text{action}[S_m, a_i]$ خالی باشد خطای نحوی رخ داده است.

M.Bazargani 93-94-2

۱۳۶

جدول تجزیه روش LR(0)

برای تشریح روش LR(0) عنصر LR را معرفی میکنیم:
عنصر LR یک قاعده تولید است که نقطه ای در سمت راست آن قرار دارد.

$$S \rightarrow \alpha.\beta$$

(مثال)
 $S \rightarrow \alpha.\beta$ نشان میدهد که برای کاهش S نیاز به کاهش α است زیرا علامت نقطه قبل از α قرار دارد.
در نتیجه برای کاهش S فعل پیشرفتی نداریم.

$$S \rightarrow \alpha.\beta$$
$$S \rightarrow \alpha\beta.$$

M.Bazargani 93-94-2

۱۳۷

مثال

LR قاعده مقابل را بنویسید.

$$S \rightarrow XYZ$$

$$S \rightarrow .XYZ$$

$$S \rightarrow X.YZ$$

$$S \rightarrow XY.Z$$

$$S \rightarrow XYZ.$$

M.Bazargani 93-94-2

۱۳۸

عناصر LR

(۱) عنصر کاهش (Reduce):

عنصر کاهش، عنصری است که نقطه در آخر سمت راست قاعده تولید قرار دارد.

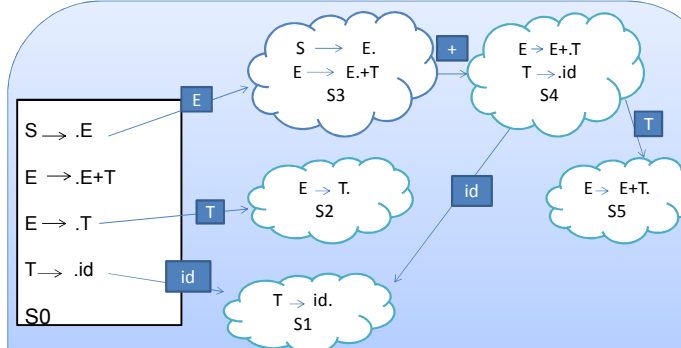
(۲) عنصر انتقالی (Shift):

عنصر انتقالی عنصری است که کاهش نباشد.
هر عنصر انتقالی نشان دهنده ی یک فرضیه در مورد دستگیره میباشد.

M.Bazargani 93-94-2

۱۳۹

مثال



M.Bazargani 93-94-2

۱۴۰

مثال

❖ تمام عناصر این مجموعه به عنصر کاهشی رسیده اند، لذا هیچ حالت دیگری به این مجموعه افزوده نخواهد شد.

ماشین خودکار قطعی به دست آمده دقیقاً یک DFA است که با استفاده از این ماشین خودکار قطعی بدست آمده میتوان جدول تجزیه را به دست آورد.

M.Bazargani ۹۳-۹۴-2

۱۴۱

مثال

حالات	Action			GOTO	
	id	+	\$	E	T
0	s1	e	e	3	2
1	r4	r4	r4		
2	r3	r3	r3		
3	e	s4	Accept		
4	s1	e	e		5
5	r2	r2	r2		

M.Bazargani ۹۳-۹۴-2

۱۴۲

مثال

توضیح سطر S0:
باتوجه به ماشین خودکار همه ی عناصر در حالت انتقالی هستند.
نمادهای E, T, id بلافاصله بعد از علامت نقطه قرار دارند که باعث تغییر حالت از حالت S0 میشوند.
نماد id باعث تغییر حالت از S0 به S1 میشود، و عنصر انتقالی id. T به عنصر کاهشی id. T تبدیل میشود و این به معنای آن است که id به بالای پشته انتقال می یابد.

Action [0, id] = shift 1 = s1

M.Bazargani ۹۳-۹۴-2

۱۴۳

مثال

توضیح سطر S1:
این حالت شامل عنصر کاهشی id $\rightarrow T$ میباشد، در این حالت یک دستگیره یافت شده است، در نتیجه در این حالت باتوجه به قاعده id $\rightarrow T$ کاهش انجام میشود.
بنابر این قطعه در S1 جدول تجزیه کاهش خواهیم داشت.

Action [1, id] = Reduce 4 = r4

M.Bazargani ۹۳-۹۴-2

۱۴۴

مثال

سطر 2 :
Action [2,\$] = r3
Action [2,id] = r3
Action [2,+] = r3
سطر 3 :
Action [3,\$] = Accept
Action [3,+] = shift 4
سطر 4 :
Action [4,id] = shift 1
سطر 5 :
Action [5,(id,+, \$)] = Reduce

M.Bazargani ۹۳-۹۴-2

۱۴۵

مثال

جدول تجزیه SLR(1) گرامر زیر را بدست آورید.

1,2 $E \rightarrow E+T \mid T$
3,4 $T \rightarrow T * F \mid F$
5,6 $F \rightarrow (E) \mid id$

M.Bazargani ۹۳-۹۴-2

۱۴۶

مثال

	id	+	*	(	)	\$	E	T	F
0	S5			S4			1	2	3
1		S6				acc			
2		r2	S7		r2	r2			
3		r4	r4		r4	r4			
4	S5			S4			8	2	3
5		r6	r6		r6	r6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		r1	S7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

M.Bazargani ۹۳-۹۴-2

۱۴۷

نکته

S: shift بعد از هر شیفت یک شماره حالت قرار دارد.
r: reduce بعد از هر reduce یک شماره قاعده قرار دارد.

M.Bazargani ۹۳-۹۴-2

۱۴۸

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$
0 E 1 + 6 T 9 * 7 F 10	$\$$	Reduce $T \rightarrow T * F$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$
0 E 1 + 6 T 9 * 7 F 10	$\$$	Reduce $T \rightarrow T * F$
0 E 1 + 6 T 9	$\$$	Reduce $E \rightarrow E + T$

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$
0 E 1 + 6 T 9 * 7 F 10	$\$$	Reduce $T \rightarrow T * F$
0 E 1 + 6 T 9	$\$$	Reduce $E \rightarrow E + T$
0 E 1	$\$$	Accept

M.Bazargani ۹۳-۹۴-۲

۱۶۲

تشخیص دستگیره ها در تجزیه گر های LR

❖ اگر فرض کنیم $S \rightarrow \alpha\beta w$ (S نماد شروع گرامر $\alpha\beta w$ یک فرم جمله راست است)

❖ اگر $\alpha\beta$ را γ بنامیم آنگاه آن را یک پیشوند قابل وقوع می نامند. در تجزیه گرهای LR از یک نمودار انتقال یا DFA استفاده میشود تا تمام پیشوندهای قابل وقوع را شناسایی کنند.

❖ اگر فرض شود در طی یک قدم کاهش به αw رسیده باشیم در این صورت β یک دستگیره است.

M.Bazargani ۹۳-۹۴-۲

۱۶۳

ساخت جدول تجزیه $SLR(1)$

تعریف قلم $LR(0)$: هر یک از قواعد گرامر که در سمت راست علامت نقطه وجود دارد.

$$A \rightarrow X_1 X_2 \dots X_i \cdot X_{i+1} \dots X_n$$

موقعیت نقطه در این قلم $LR(0)$ مشخص میکند که در بالای پشته قرار دارد و انتظار می رود که نیز دیده شود.

مثال: اقلام $LR(0)$ قاعده $A \rightarrow XYZ$ بصورت زیر است:

$$A \rightarrow \cdot XYZ$$

$$A \rightarrow X \cdot YZ$$

$$A \rightarrow XY \cdot Z$$

$$A \rightarrow XYZ \cdot$$

M.Bazargani ۹۳-۹۴-۲

۱۶۴

ساخت جدول تجزیه SLR(1)

نکته: اگر طول سمت راست یک قاعده برابر n باشد آنگاه میتوان $n+1$ قلم LR(0) متمایز برای آن قاعده داشت.
تعریف تابع بستار: ورودی تابع بستار مجموعه ای از اقلام LR(0) است که بصورت زیر محاسبه میشود:
ابتدا اقلام LR(0) مجموعه ورودی را در خروجی می گذاریم و در خروجی دنبال قلم LR(0) به شکل $A \rightarrow \alpha.B\beta$ می گردیم بطوری که علامت نقطه قبل از غیرپایانه B باشد.

M.Bazargani ۹۳-۹۴-۲

۱۶۵

ساخت جدول تجزیه SLR(1)

تعریف تابع بستار: ورودی و خروجی تابع بستار مجموعه ای از اقلام LR(0) بوده که به صورت زیر محاسبه می شود:
ابتدا اقلام LR(0) مجموعه ورودی را در خروجی می گذاریم و در خروجی دنبال قلم LR(0) به شکل $A \rightarrow \alpha.B\beta$ می گردیم به طوری که علامت نقطه قبل از غیر پایانه B باشد تمامی قواعد غیرپایانه B که به شکل $B \rightarrow \gamma$ هستند را در نظر می گیریم و اقلام LR(0) جدید $B \rightarrow \gamma$ را به خروجی اضافه می کنیم و این عمل را ادامه می دهیم تا هیچ قلم LR(0) جدیدی به خروجی اضافه نشود. اگر برای غیرپایانه B قاعده $B \rightarrow \lambda$ داشتیم قلم LR(0) $B \rightarrow \lambda$ را به خروجی اضافه می کنیم.

M.Bazargani 93-94-2

۱۶۶

مثال

بستار گرامر زیر را محاسبه کنید:

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

M.Bazargani ۹۳-۹۴-۲

۱۶۷

مثال

$I_1 = \{E \rightarrow .E+T\}$
 $\text{closure}(I_1) = \{E \rightarrow .E+T, E \rightarrow .T, T \rightarrow .T * F, T \rightarrow .F, F \rightarrow .(E), F \rightarrow .id\}$
 $I_2 = \{E \rightarrow .E+T, T \rightarrow .F\}$
 $\text{closure}(I_2) = \{E \rightarrow .E+T, T \rightarrow .F, F \rightarrow .(E), F \rightarrow .id\}$
 $I_3 = \{T \rightarrow .T * F\}$
 $\text{closure}(I_3) = I_3$

M.Bazargani ۹۳-۹۴-۲

۱۶۸

رسم نمودار انتقال SLR(1)

❖ برای ساخت جدول تجزیه SLR(1) باید ابتدا نمودار انتقال SLR(1) رسم شود این نمودار انتقال یک DFA است و برای تشخیص تمامی پیشوند های قابل وقوع استفاده می شود. با فرض این که نماد شروع گرامر غیر پایانه S باشد در صورتی که در گرامر قاعده افزوده $S' \rightarrow S$ وجود نداشته باشد این قاعده به گرامر اضافه می شود به گرامر معادل جدید گرامر افزوده گویند.



M.Bazargani ۹۳-۹۴-2

۱۶۹

رسم نمودار انتقال SLR(1)

ابتدا در حالت صفر قلم LR(0), $S \rightarrow S'$ را قرار می دهیم و سپس بستارش را به این حالت اضافه می کنیم تا به این ترتیب حالت صفر تشکیل شود سپس در حالت i دنبال اقلام LR(0) به شکل زیر می گردیم:

$$A_1 \rightarrow \alpha_1 \cdot x \beta_1$$

$$A_2 \rightarrow \alpha_2 \cdot x \beta_2$$

.....

$$A_n \rightarrow \alpha_n \cdot x \beta_n$$

M.Bazargani ۹۳-۹۴-2

۱۷۰

مثال

نمودار انتقال SLR(1) گرامر زیر را رسم کنید.

1,2 $E \rightarrow E+T \mid T$

3,4 $T \rightarrow T * F \mid F$

5,6 $F \rightarrow (E) \mid id$

M.Bazargani ۹۳-۹۴-2

۱۷۱

ساخت جدول تجزیه SLR(1)

❖ ساخت جدول تجزیه SLR(1) از روی نمودار انتقال SLR(1):

❖ جدول تجزیه از دو بخش action و goto تشکیل شده است.

❖ برای تکمیل بخش action به صورت زیر عمل می کنیم:

M.Bazargani ۹۳-۹۴-2

۱۷۲

ساخت جدول تجزیه SLR(1)

- ❖ پر کردن انتقال ها: اگر در حالتی مانند i با پایانه a به حالتی مانند j رفتیم آنگاه در بخش action سطر i و زیر ستون پایانه a دستور انتقال و به دنبال آن شماره حالت j را قرار می دهیم.
- ❖ پر کردن کاهش ها: اگر در حالتی مانند A علامت نقطه در انتهای سمت راست قاعده به شکل $A \rightarrow \alpha$ باشد نشان دهنده این است که بالای پشته دستگیره α وجود دارد و می توان عمل کاهش به غیر پایانه A صورت گیرد پس در سطر i از بخش action و زیر تمامی عناصر $follow(A)$ دستور کاهش و به دنبال آن شماره قاعده $A \rightarrow \alpha$ را قرار می دهیم.
- ❖ فقط در سطر حالتی که قلم $LR(0)$, $S' \rightarrow S$ قرار دارد و زیر ستون $\$$ از بخش action کلمه accept را قرار می دهیم.

M.Bazargani ۹۳-۹۴-2

۱۷۳

ساخت جدول تجزیه SLR(1)

- ❖ برای تکمیل بخش goto به صورت زیر عمل می کنیم:
- ❖ اگر در حالتی مانند i با غیر پایانه A به حالتی مانند j رفتیم آنگاه در سطر i از بخش goto و زیر ستون غیر پایانه A شماره حالت j را قرار می دهیم.

M.Bazargani ۹۳-۹۴-2

۱۷۴

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift

۱۷۵

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce F $\rightarrow id$

۱۷۶

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$

۱۷۷

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$

۱۷۸

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift

۱۷۹

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift

۱۸۰

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$

۱۸۱

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$

۱۸۲

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift

۱۸۳

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift

۱۸۴

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$

۱۸۵

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$

۱۸۶

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$
0 E 1 + 6 T 9 * 7 F 10	$\$$	Reduce $T \rightarrow T * F$

۱۸۷

مثال: رشته $id+id*id$ را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	$id + id * id \$$	Shift
0 id 5	$+ id * id \$$	Reduce $F \rightarrow id$
0 F 3	$+ id * id \$$	Reduce $T \rightarrow F$
0 T 2	$+ id * id \$$	Reduce $E \rightarrow T$
0 E 1	$+ id * id \$$	Shift
0 E 1 + 6	$id * id \$$	Shift
0 E 1 + 6 id 5	$* id \$$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	$* id \$$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	$* id \$$	Shift
0 E 1 + 6 T 9 * 7	$id \$$	Shift
0 E 1 + 6 T 9 * 7 id 5	$\$$	Reduce $F \rightarrow id$
0 E 1 + 6 T 9 * 7 F 10	$\$$	Reduce $T \rightarrow T * F$
0 E 1 + 6 T 9	$\$$	Reduce $E \rightarrow E + T$

۱۸۸

مثال: رشته id+id*id را با استفاده از جدول تجزیه فوق تجزیه کنید.

پشته	بخش باقیمانده ورودی	عمل
0	id + id * id \$	Shift
0 id 5	+ id * id \$	Reduce $F \rightarrow id$
0 F 3	+ id * id \$	Reduce $T \rightarrow F$
0 T 2	+ id * id \$	Reduce $E \rightarrow T$
0 E 1	+ id * id \$	Shift
0 E 1 + 6	id * id \$	Shift
0 E 1 + 6 id 5	* id \$	Reduce $F \rightarrow id$
0 E 1 + 6 F 3	* id \$	Reduce $T \rightarrow F$
0 E 1 + 6 T 9	* id \$	Shift
0 E 1 + 6 T 9 * 7	id \$	Shift
0 E 1 + 6 T 9 * 7 id 5	\$	Reduce $F \rightarrow id$
0 E 1 + 6 T 9 * 7 F 10	\$	Reduce $T \rightarrow T * F$
0 E 1 + 6 T 9	\$	Reduce $E \rightarrow E + T$
0 E 1	\$	Accept

M.Bazargani 93-94-2

۱۸۹

ساخت جدول تجزیه SLR(1)

❖ نکته: اگر هیچ تداخلی در خانه های بخش action وجود نداشته باشد گرامر داده شده یک گرامر SLR(1) خواهد بود.

❖ نکته: شرط لازم برای اینکه تداخل s/r در جدول تجزیه وجود داشته باشد این است که حالتی در نمودار انتقال باشد که در آن هم دستگیره و هم انتقال داشته باشیم.

❖ نکته: شرط لازم برای وجود تداخل r/r این است که حالتی در نمودار انتقال داشته باشیم که در آن بیش از یک دستگیره باشد.

M.Bazargani ۹۳-۹۴-۲

۱۹۰

مثال

مثال: جدول تجزیه SLR(1) گرامر زیر را به دست آورید.

- 1 $S \rightarrow L = R$
- 2 $S \rightarrow R$
- 3 $L \rightarrow * R$
- 4 $L \rightarrow id$
- 5 $R \rightarrow L$

M.Bazargani ۹۳-۹۴-۲

۱۹۱

مثال

	id	=	*	\$	S	L	R
0	S5		S4		1	2	3
1				acc			
2		r5,S6		r5			
3				r2			
4	S5		S4			8	7
5		r4		r4			
6	S5		S4			8	9
7		r3		r3			
8		r5		r5			
9				r1			

M.Bazargani ۹۳-۹۴-۲

۱۹۲

تعریف قلم هسته ای

❖ به قلم $LR(0)$ ، $S' \rightarrow S$ و هر قلم $LR(0)$ که نقطه در ابتدای سمت راست آن نباشد یک قلم هسته ای می گویند. وقتی نمودار انتقال خیلی بزرگ باشد برای نگهداری آن حافظه ی زیادی مصرف می شود . برای صرفه جویی در مصرف حافظه می توان در هر حالت فقط اقلام هسته ای آن حالت را نگهداری کرد و هر وقت نیاز باشد اقلام غیر هسته ای را به سادگی با اعمال تابع بستار روی اقلام هسته ای بدست آورد.

M.Bazargani 93-94-2

۱۹۳

گرامر های $LR(0)$ و نحوه ساخت جدول تجزیه $LR(0)$

❖ برای اینکه جدول تجزیه $LR(0)$ را ایجاد کنیم ابتدا باید نمودار انتقال $LR(0)$ را رسم کنیم. نمودار انتقال $LR(0)$ دقیقاً همان نمودار انتقال $SLR(1)$ است . پس برای ساخت جدول تجزیه $LR(0)$ ابتدا باید نمودار انتقال $SLR(1)$ ساخته شود و از روی آن بر اساس قدم های زیر جدول انتقال $LR(0)$ تشکیل شود:

M.Bazargani 93-94-2

۱۹۴

گرامر های $LR(0)$ و نحوه ساخت جدول تجزیه $LR(0)$

❖ تکمیل بخش action

- پر کردن انتقال ها: شبیه پر کردن انتقال های بخش action در جدول تجزیه $SLR(1)$ است.
- پر کردن کاهش ها: اگر در حالتی مانند i علامت نقطه در انتهای سمت راست قلم $LR(0)$ به شکل $A \rightarrow \alpha$ باشد دستگیره یافت شده و در این حالت عمل کاهش داریم. در سطر i بخش action و زیر ستون های تمامی پایانه ها و همچنین علامت $\$$ دستور کاهش و به دنبال آن شماره قاعده $A \rightarrow \alpha$ قرار داده می شود.
- در سطر حالتی که قلم $LR(0)$ ، $S' \rightarrow S$ قرار دارد و زیر ستون $\$$ کلمه accept قرار داده می شود.

M.Bazargani 93-94-2

۱۹۵

گرامر های $LR(0)$ و نحوه ساخت جدول تجزیه $LR(0)$

تکمیل بخش goto:

- خانه های بخش goto شبیه بخش goto جدول تجزیه $SLR(1)$ تکمیل می شود.
- نکته: شرط لازم و کافی برای وجود تداخل s/r در یک حالت از نمودار انتقال $LR(0)$ این است که در آن حالت هم انتقال و هم کاهش وجود داشته باشد.
- نکته: شرط لازم و کافی برای وجود تداخل r/r در یک حالت از نمودار انتقال $LR(0)$ این است که در آن حالت بیش از یک دستگیره وجود داشته باشد.

M.Bazargani 93-94-2

۱۹۶

مثال

جدول تجزیه LR(0) گرامر زیر را بدست آورید.

- 1 $S \rightarrow aS$
- 2 $S \rightarrow bA$
- 3 $A \rightarrow d$

M.Bazargani 93-94-2

۱۹۷

مثال

	a	b	d	\$	S	A
0	S2	S3			1	
1				acc		
2	S2	S3			4	
3			S6			5
4	r1	r1	r1	r1		
5	r2	r2	r2	r2		
6	r3	r3	r3	r3		

M.Bazargani 93-94-2

۱۹۸

ساخت جدول تجزیه CLR(1)

- ❖ ابتدا نمودار انتقال CLR(1) باید رسم شود.
- ❖ تعریف قلم LR(1): یک زوج مرتب است که مولفه اول آن یک قلم LR(0) بوده و مولفه دوم آن مجموعه ای از پایانه ها بنام مجموعه پیشنگر است.
- ❖ یک قلم LR(1) به شکل $[A \rightarrow \alpha \cdot \beta, LA]$ نشان داده میشود. همان مجموعه پیشبینی است و همواره LA زیر مجموعه follow(A) است.
- ❖ در روش تجزیه CLR(1) به جای قلم LR(0) از قلم LR(1) استفاده می شود. بنابر این وقتی نقطه در انتهای سمت راست مولفه اول یک قلم LR(1) به شکل $[A \rightarrow \alpha \cdot, LA]$ قرار گیرد فقط کاهش به ازای مجموعه پایانه های LA خواهیم داشت نه تمامی عناصر follow(A).

M.Bazargani 93-94-2

۱۹۹

ساخت جدول تجزیه CLR(1)

تعریف تابع بستار: ورودی این تابع مجموعه ای از اقلام LR(1) است که بصورت زیر محاسبه می شود:

ابتدا تمامی اقلام LR(1) ورودی را در مجموعه خروجی قرار می دهیم و در مجموعه خروجی دنبال قلم LR(1) به شکل $[A \rightarrow \alpha \cdot B\beta, \{a_1, a_2, \dots, a_n\}]$ میگردیم. به طوری که علامت نقطه قبل از غیر پایانه باشد سپس قواعد غیر پایانه را که به شکل $B \rightarrow \gamma$ هستند در نظر می گیریم و تمامی اقلام LR(1) جدید $[B \rightarrow \gamma \cdot, \{b_1, b_2, \dots, b_m\}]$ را به خروجی اضافه می کنیم در حالی که :

M.Bazargani 93-94-2

۲۰۰

ساخت جدول تجزیه CLR(1)

$$\{b_1, b_2, \dots, b_n\} = \text{first}(\beta a_1) \cup \text{first}(\beta a_2) \cup \dots \cup \text{first}(\beta a_n)$$

این عمل را ادامه می دهیم تا هیچ قلم LR(1) جدیدی به مجموعه خروجی اضافه نشود.

M.Bazargani 93-94-2

۲.۱

رسم نمودار انتقال CLR(1)

در صورتی که قاعده افزوده $S' \rightarrow S$ در گرامر وجود نداشته باشد این قاعده را به گرامر اضافه می کنیم. در حالت صفر قلم LR(1), $[S' \rightarrow S, \{\$ \}]$ را قرار می دهیم و بستارش را به حالت صفر اضافه می کنیم تا حالت صفر تشکیل شود و شبیه روش رسم نمودار انتقال SLR(1) سایر حالت را بدست می آوریم با این تفاوت که در حالت های نمودار انتقال CLR(1) اقلام LR(1) را خواهیم داشت.

M.Bazargani 93-94-2

۲.۲

مثال

نمودار انتقال CLR(1) گرامر زیر را رسم کنید.

- 1 $S \rightarrow GG$
- 2 $G \rightarrow cG$
- 3 $G \rightarrow d$

نکته: از رفتن از یک حالت به حالت دیگر مجموعه پیشبینی عوض نمی شود و فقط موقعیت علامت نقطه عوض می شود

M.Bazargani 93-94-2

۲.۳

ساخت جدول تجزیه CLR(1)

ساخت جدول تجزیه CLR(1) از روی نمودار انتقال CLR(1) بسیار مشابه روش ساخت جدول انتقال SLR(1) از روی نمودار انتقال SLR(1) است.

تکمیل بخش goto: جدول تجزیه CLR(1) مشابه پر کردن خانه های بخش goto جدول SLR(1) است.

تکمیل بخش action: پر کردن انتقال ها مشابه بخش goto جدول تجزیه SLR(1) است. تفاوت فقط در کاهش ها است.

M.Bazargani 93-94-2

۲.۴

ساخت جدول تجزیه CLR(1)

اگر در حالتی مانند i قلم LR(1) به شکل $[A \rightarrow \alpha \cdot, LA]$ داشته باشیم آنگاه در سطر i از بخش action و زیر پایانه های مجموعه پیشبینی LA دستور کاهش و به دنبال آن شماره قاعده $A \rightarrow \alpha$ را قرار می دهیم همچنین در سطر حالتی که قلم LR(1)، $[S' \rightarrow S \cdot, \{\$ \}]$ قرار دارد و زیر ستون \$ کلمه accept را قرار می دهیم.

جدول زیر جدول تجزیه CLR(1) گرامر فوق را نشان می دهد:

M.Bazargani 93-94-2

۲۰۵

مثال

	c	d	\$	s	G
0	S3	S4		1	2
1			acc		
2	S6	S7			5
3	S3	S4			8
4	r3	r3			
5			r1		
6	S6	S7			9
7			r3		
8	r2	r2			
9			r2		

M.Bazargani 93-94-2

۲۰۶

ساخت جدول تجزیه LALR(1)

برای ساخت این جدول باید نمودار انتقال LALR(1) رسم شود اما برای رسم نمودار انتقال LALR(1) ابتدا باید نمودار انتقال CLR(1) رسم شود و آن را به نمودار انتقال LALR(1) تبدیل کرد:

اگر تمامی مولفه های اول اقلام LR(1) یک حالت یعنی اقلام LR(0) را هسته آن حالت بنامیم در نمودار انتقال CLR(1) حالت های با هسته یکسان را می یابیم و آن ها را با یکدیگر ادغام می کنیم. عمل ادغام به این صورت انجام می شود که در حالت ادغام شده هسته را می نویسیم و مولفه دوم، یعنی مجموعه پیش بینی از اجتماع مجموعه های پیش بینی حالت های قابل ادغام به دست می آید با ادغام تمام حالت های با هسته یکسان نمودار انتقال LALR(1) بدست می آید.

M.Bazargani 93-94-2

۲۰۷

مثال

ساخت جدول تجزیه LALR(1) از روی نمودار انتقال LALR(1) دقیقاً مشابه ساخت جدول تجزیه CLR(1) از روی نمودار انتقال CLR(1) است.

مثال: جدول تجزیه CLR(1), LALR(1), SLR(1) گرامر زیر را بدست آورید.

- 0 $S' \rightarrow S$
- 1 $S \rightarrow V := E$
- 2 $S \rightarrow id$
- 3 $V \rightarrow id$
- 4 $E \rightarrow V$
- 5 $E \rightarrow num$

M.Bazargani 93-94-2

۲۰۸

مثال

	c	d	\$	s	G
0	S3-6	S4-7		1	2
1			acc		
2	S3-6	S4-7			5
3-6	S3-6	S4-7			8-9
4-7	r3	r3	r3		
5			r1		
8-9	r2	r2	r2		

M.Bazargani 93-94-2

۲۰۹

مزیت استفاده از گرامرهای مبهم در روش تجزیه LR

گرامر G_1 زیر یک گرامر مبهم است

1,2,3 $E \rightarrow E+E \mid E^*E \mid id$

گرامر غیر مبهم G_1 :

G_2 :
 $E \rightarrow E+T \mid T$
 $T \rightarrow T^*F \mid F$
 $F \rightarrow id$

M.Bazargani 93-94-2

۲۱۰

مثال

اگر رشته $id+id$ را توسط G_1 تجزیه کنیم تعداد مراحل اشتقاق ۳ است:

$E \Rightarrow E+E \Rightarrow E + id \Rightarrow id + id$

اگر رشته $id+id$ را توسط G_2 تجزیه کنیم تعداد مراحل اشتقاق ۶ است.

$E \Rightarrow E+T \Rightarrow E+F \Rightarrow E+id \Rightarrow T+id \Rightarrow F+id \Rightarrow id+id$

M.Bazargani 93-94-2

۲۱۱

مثال

	id	+	*	\$	E
0	S2				1
1		S3	S4	acc	
2		r3	r3	r3	
3	S2				5
4	S2				6
5		r1,S3	r1, S4	r1	
6		S3,r2	r2,S4	r2	

M.Bazargani 93-94-2

۲۱۲

مثال

❖ همانطور که ملاحظه می شود ۴ تداخل در جدول تجزیه (SLR(1 گرامر مبهم G_1 وجود دارد به جای اینکه از گرامر غیر مبهم معادل استفاده کنیم میتوان از خود گرامر مبهم استفاده کرد و مشکل ابهام را در جدول تجزیه حل کرد بدین صورت که در جدول تداخل هارا رفع کنیم.

❖ در هر خانه که تداخل وجود دارد با توجه به اولویت عملگرها عمل درست را نگه داشته و عمل نادرست را حذف کنیم.

❖ به عنوان مثال:

M.Bazargani 93-94-2

۲۱۳

مثال

❖ خانه $action[5,+]$ برابر اعمال S_3 و r_1 است یعنی در حالت ۵ نمودار انتقال تداخل s/r وجود دارد. با نگاه کردن به حالت ۵ پی برده می شود که بالای پشته دستگیره $E+E$ وجود دارد با فرض اینکه توکن جاری + باشد اگر عمل کاهش انجام شود اولویت جمع سمت چپ بیشتر می شود و اگر عمل انتقال انجام شود اولویت جمع سمت راست بیشتر است. جمع دارای خاصیت شرکت پذیری چپ است پس عمل S_3 باید از این خانه حذف شود همچنین در خانه $action[5,*]$ تداخل وجود دارد و برای اینکه اولویت ضرب از جمع بیشتر شود باید از این خانه r_1 حذف شود.

M.Bazargani 93-94-2

۲۱۴

راه حل

برای برخورد با گرامر مبهم دو راه وجود دارد:

✓ استفاده از خود گرامر مبهم و ایجاد جدول تجزیه LR و رفع تداخل در جدول از طریق حذف عمل درست و حذف عمل نادرست.

✓ تبدیل گرامر مبهم به یک گرامر غیر مبهم و طراحی تجزیه گر LR برای گرامر غیر مبهم معادل.

M.Bazargani 93-94-2

۲۱۵

رابطه بین گرامرها و زبان های $LL(k)$ و $LR(k)$

مثال: بررسی کنید گرامر زیر $SLR(1)$ است یا $LALR(1)$ است؟

 $S \rightarrow AaAb$ $S \rightarrow BbBa$ $A \rightarrow \lambda$ $B \rightarrow \lambda$

	a	b	\$
S	1	2	
A	3	3	
B	4	4	

گرامر فوق $LL(1)$ است زیرا شرایط آن را داراست.

M.Bazargani 93-94-2

۲۱۶

رابطه بین گرامرها و زبان های LL(k) و LR(k)

❖ اگر به حالت صفر نمودار توجه شود در این حالت دو دستگیره وجود دارد و اشتراک $\text{follow}(A)$ و $\text{follow}(B)$ تهی نیست پس در این حالت تداخل r/r وجود دارد و گرامر داده شده $\text{SLR}(1)$ نیست.

❖ اگر به حالت صفر نمودار $\text{CLR}(1)$ توجه شود در این حالت دو دستگیره وجود دارد و اشتراک مجموعه های پیش بینی آنها تهی است.

M.Bazargani 93-94-2

۲۱۷

نکات

نکته: اگر گرامری $\text{CLR}(1)$ نباشد حتماً $\text{LL}(1)$ و $\text{LALR}(1)$ و $\text{SLR}(1)$ نیز نیست.

نکته: هر گرامر $\text{LL}(1)$ لزوماً $\text{LALR}(1)$ نیست.

نکته: هر گرامر $\text{LALR}(1)$ یک گرامر $\text{CLR}(1)$ است ولی عکس آن لزوماً درست نیست.

نکته: هر گرامر $\text{LL}(1)$ لزوماً $\text{SLR}(1)$ نیست.

M.Bazargani 93-94-2

۲۱۸

رفع خطای نحوی در تجزیه گره های LR

در روش تجزیه LR وقتی خطای نحوی رخ می دهد که تجزیه گر برای انتخاب عمل صحیح به بخش action جدول تجزیه مراجعه کند و خانه مربوطه خالی باشد.

M.Bazargani 93-94-2

۲۱۹

روش رفع خطای panic mode

اگر فرض کنید در طی عمل تجزیه تجزیه گر در وضعیت زیر باشد :
 $\langle S_0 X_1 S_1 X_2 S_2 \dots X_k S_k \dots X_m S_m, a_{i+1} \dots a_{j+1} \dots a_n \rangle$
 و $\text{action}[S_m, a_i]$ خالی باشد تجزیه گر با خطای نحوی مواجه شده است.

بنابراین باید از بالای پشته تعدادی علامت را حذف کنیم و همچنین تعدادی از توکن های ورودی را باید نادیده بگیریم تا بتوان ادعا کرد یک دستگیره یافت شده است و باید آن را به یک غیره پایانه مانند A کاهش داد .

M.Bazargani 93-94-2

۲۲۰

روش رفع خطای panic mode

با این کار تجزیه گراز روی خطا رد می شود و به کار تجزیه ادامه می دهد. در این روش رفع خطا، تلاش میشود جمله حاوی خطای نحوی به کلی نادیده گرفته شود. به عبارت دیگر با انتخاب غیر پایانه ی مانند A مشخص کرده ایم که رشته قابل اشتقاق از A حاوی خطای نحوی است. البته بخشی از این رشته قبلا پردازش شده که در بالای پشته است و باقیمانده رشته هنوز در ورودی قرار دارد. چه تعدادی از نمادهای بالای پشته و ورودی را دور بریزیم میتواند بدین شکل باشد که عناصر بالای پشته را حذف کرده تا به حالتی مانند S_k برسیم بطوری که در حالت S_k پرش به غیر پایانه ی مانند A داشته باشیم یعنی $goto[S_k, A]$ برابر حالتی مانند S_1 باشد.

M.Bazargani 93-94-2

۲۲۱

روش رفع خطای panic mode

سپس توکن های ورودی را نادیده می گیریم تا به توکنی مانند a_j برسیم در حالی که عضو مجموعه $follow(A)$ باشد. حال ابتدا A را وارد پشته می کنیم و روی آن را قرار می دهیم و به این ترتیب تجزیه گر در وضعیت جدید زیر قرار خواهد گرفت.

$$\langle S_0 X_1 S_1 X_2 S_2 \dots X_k S_k A S_{i+1} a_{j+1} \dots a_n \rangle$$

به این ترتیب تجزیه گر با حذف علامات بالای پشته، کنار گذاشتن بخشی از ورودی، قرار دادن A بالای پشته و قرار دادن $goto[S_k, A]$ روی آن، تظاهر می کند که دستگیره ای را یافته و قادر است به فرایند تجزیه ادامه دهد.

M.Bazargani 93-94-2

۲۲۲

روش اصلاح خطای phrase level

همانطور که قبلا اشاره شد این روش با حذف و اضافه کردن یک یا چند نماد به پشته و یا حذف و اضافه کردن یک یا چند توکن به ورودی سعی در اصلاح خطا دارد. پس این روش، خطا را در محل وقوع اصلاح می کند. با توجه به گرامر و زبان آن، در این روش برای هر خانه ی خالی بخش action جدول تجزیه که نشان دهنده خطای نحوی است یک روال مناسب اصلاح خطا می نویسیم و آدرس روال را در آن خانه خالی قرار میدهم. وقتی تجزیه گر با خطا مواجه می شود روال مربوطه را فراخوانی می کند تا خطا اصلاح شود و تجزیه گر بتواند به کارش ادامه دهد.

M.Bazargani 93-94-2

۲۲۳

مثال

روال های اصلاح خطا را برای جدول تجزیه ی $SLR(1)$ گرامر زیر بنویسید.

$$1,2,3,4 \quad E \rightarrow E+E \mid E^*E \mid (E) \mid id$$

M.Bazargani 93-94-2

۲۲۴

مثال

	id	+	*	(	)	\$	E
0	S3			S2			1
1		S4	S5			acc	
2	S3			S2			6
3		r4	r4		r4	r4	
4	S3			S2			7
5	S3			S2			8
6		S4	S5		S9		
7		r1	S5		r1	r1	
8		r2	r2		r2	r2	
9		r3	r3		r3	r3	

M.Bazargani 93-94-2

۲۲۵

جدول تجزیه (1)SLR با اضافه کردن روال های اصلاح خطا

	id	+	*	(	)	\$	E
0	S3	e1	e1	S2	e2	e1	1
1	e3	S4	S5	e3	e2	acc	
2	S3	e1	e1	S2	e2	e1	6
3	r4	r4	r4	r4	r4	r4	
4	S3	e1	e1	S2	e2	e1	7
5	S3	e1	e1	S2	e2	e1	8
6	e3	S4	S5	e3	S9	e4	
7	r1	r1	S5	r1	r1	r1	
8	r2	r2	r2	r2	r2	r2	
9	r3	r3	r3	r3	r3	r3	

M.Bazargani 93-94-2

۲۲۶

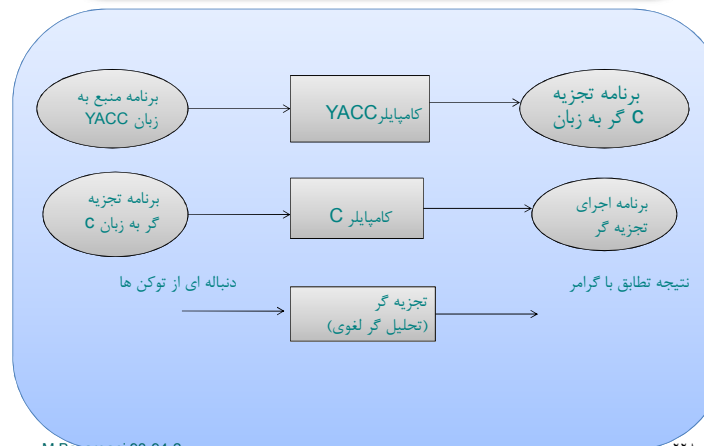
تولید خودکار تجزیه گر با استفاده از ابزار

معمولا تجزیه گرها را بطور خودکار با استفاده از ابزار تولید می کنند. یکی از ابزار های معروف برای تولید خودکار تجزیه گر ابزار YACC است. ابزار YACC برای ایجاد خودکار برنامه تجزیه گر از روش تجزیه ی (1)LALR استفاده می کند. برای تولید خودکار برنامه ی تجزیه گر میبایست یک گرامر به عنوان ورودی به ابزار YACC داده شود تا بر اساس آن تجزیه گر مورد نظر را تولید کند.

M.Bazargani 93-94-2

۲۲۷

مراحل تولید خودکار تجزیه گر با استفاده از ابزار YACC



M.Bazargani 93-94-2

۲۲۸

مراحل تولید خودکار تجزیه گر با استفاده از ابزار YACC

برنامه ی منبع به زبان YACC باید در یک فایل با پسوند y. نوشته شود.
ساختار برنامه ی ورودی ابزار YACC باید به صورت زیر باشد:

```
%{
    اعلان متغیر ها
    اعلان ثوابت
}%
اعلان توکن های موجود در گرامر
%%
قواعد گرامر
%%
روال های پشتیبان به زبان C
```

M.Bazargani 93-94-2

۲۲۹

مراحل تولید خودکار تجزیه گر با استفاده از ابزار YACC

ساختار فوق مشخص می کند برنامه به زبان YACC از سه بخش مجزا تشکیل شده است که توسط %/ از هم جدا شده اند. بخش اول شامل اعلان متغیرها، ثوابت و همچنین توکن های موجود در گرامر است. بخش دوم شامل تعریف قواعد تولید موجود در گرامر است. شکل کلی تعریف قواعد یک گرامر به صورت زیر است :

<انتخاب n>|...|<انتخاب ۲>|<انتخاب ۱>→<سمت چپ>

M.Bazargani 93-94-2

۲۳۰

مثال

برای گرامر زیر با استفاده از کامپایلر YACC یک تجزیه گر تولید کنید.

```
S→Expr
Expr→Expr + Term | Term
Term→Term*Factor | Factor
Factor→id | num | (Expr)
```

M.Bazargani 93-94-2

۲۳۱

مثال

```
%token num
%token id
%%
S : Expr
  ;
Expr : Expr '+' Term
     | Term
     ;
Term : Term '*' Factor
     | Factor
     ;
Factor : id
       | num
       | '('Expr')'
```

M.Bazargani 93-94-2

۲۳۲

مثال

اگر جدول تجزیه ی LR را برای گرامر فوق بسازیم ۴ تداخل s/r در جدول خواهیم داشت. همچنین اگر فایل ورودی را به ابزار YACC بدهیم، ابزار YACC نیز وجود این ۴ تداخل را گزارش خواهد داد. ابزار YACC این امکان را فراهم میکند که بتوان اولویت مورد نظر را طبق نحو YACC اعلام کرد. عمل کردهای + ، - ، * و / همگی دارای شرکت پذیری چپ هستند و تقدم * و / از + و - بیشتر است بنابراین در بخش اول فایل ورودی میتوان اولویت مد نظر را بصورت زیر بیان کرد:

%left '+', '-'

%left '*', '/'

M.Bazargani 93-94-2

۲۳۳

مثال

این دو دستور بیان میکند که توکن های + ، - ، * ، / دارای خاصیت شرکت پذیری چپ هستند و همچنین از بالا به پایین تقدم بیشتر می شود به دلیل اینکه * و / از بال به پایین بهد از + و - آمده است دارای اولویت بیشتر هستند. اگر بخواهیم عملگر دارای خاصیت شرکت پذیری راست باشد از %right استفاده می کنیم.

M.Bazargani 93-94-2

۲۳۴