اصول طراحی کامپایلر

درس‌نامه‌ی ۱۳

کاظم فولادی
http://kazim.fouladi.ir

ویراست اول: ۱۳۸۸
ویراست دوم: ۱۳۹۳

فهرست مطالب

۱۳ گرامرهای خصیصه‌دار و ترجمه‌ی هدایت‌شده با نحو ۱

۱۳-۱ مقدمه ۱

۱۳-۲ ترجمه‌ی هدایت شده با نحو ۱

۱۳-۳ تعریف‌های هدایت شده با نحو ۲

۱۳-۳-۱ شکل‌های تعریف‌های هدایت شده با نحو ۳

۱۳-۳-۲ تعریف‌های S-خصیصه‌ای ۴

۱۳-۳-۳ تعریف‌های L-خصیصه‌ای ۴

خصیصه‌های موروثی ۵

۱۳-۳-۴ بازنویسی تعریف برای استفاده از خصیصه‌های ترکیبی ۷

۱۳-۳-۵ پیاده‌سازی ارزیابی خصیصه‌ها ۷

گراف‌های وابستگی ۷

ترتیب ارزیابی ۸

۱۳-۳-۶ تعریف‌های هدایت شده با نحو اکیداً بدون چرخه ۹

ارزیابی تعریف‌های S-خصیصه‌ای ۹

ارزیابی تعریف‌های L-خصیصه‌ای ۱۰

۱۳-۴ شِماهای ترجمه ۱۱

۱۳-۴-۱ طراحی شماهای ترجمه ۱۳
۱۳-۴-۲ ارزیابی شماهای ترجمه در زمان کامپایل ۱۳

❊ تمرین ۱۶

۱۳ گرامرهای خصیصه‌دار و ترجمه‌ی هدایت‌شده با نحو

ARRTIBUTE GRAMMARS AND SYNTAX DIRECTED TRANSLATION

۱-۱۳ مقدمه

- هنگامی که ساختار نحوی شناخته شد، **تحلیل معنایی** اطلاعات اضافی لازم را محاسبه‌می‌کند.
- اطلاعاتی که باید محاسبه شود، خارج از توانایی تکنیک‌های استاندارد تجزیه است (نیاز به گرامرهای سطوح بالاتر مانند گرامرهای حساس به متن). بنابراین، این اطلاعات به عنوان «نحو» در نظر گرفته نمی‌شود.
- این اطلاعات، به معنای برنامه مربوط است.
- اطلاعات مربوط به معنای برنامه معمولاً برای تکمیل محتوای جدول نمادها استفاده می‌شود.
- برای انجام بررسی‌های معنایی، به **ساز و کار بازنمایی** و **ساز و کار پیاده‌سازی** نیاز داریم.
  - **ساز و کار بازنمایی:** گرامرهای خصیصه‌دار (تعریف‌های هدایت شده با نحو)
  - **ساز و کار پیاده‌سازی:** شماهای ترجمه

۲-۱۳ ترجمه‌ی هدایت شده با نحو

> اصل **ترجمه‌ی هدایت شده با نحو**:
> معنای یک جمله‌ی ورودی وابسته به ساختار نحوی آن (یعنی درخت تجزیه) است.

ترجمه‌ی هدایت‌شده با نحو، مکانیزمی را برای ترجمه‌ی ساختارهای یک زبان که با گرامر مستقل از متن توصیف شده است، فراهم می‌کند:

- به هر نماد گرامر، **خصیصه (Attribute)**هایی نسبت داده می‌شود.
- مقادیر خصیصه‌ها توسط **قواعد معنایی (Semantic rule)** نسبت داده شده به قواعد گرامر محاسبه می‌شود.

ارزیابی قواعد معنایی می‌تواند موجب نتایج زیر شود:

- تولید کد؛
- درج اطلاعات در جدول نمادها؛
- انجام بررسی معنایی؛

- صدور پیام خطا؛
- ...

دو مفهوم در ارتباط با الصاق قواعد معنایی به قواعد گرامر وجود دارد:

- **تعریف هدایت شده با نحو (Syntax Directed Definition)**:
  توصیف سطح‌بالایی که بسیاری از جزئیات پیاده‌سازی را پنهان می‌سازد (گرامر خصیصه‌دار)
- **شماهای ترجمه (Translation Schemes)**:
  توصیفی با گرایش بیشتر به پیاده‌سازی: مشخص‌سازی ترتیب ارزیابی قواعد معنایی

مثال

برای توضیح تعریف‌های هدایت شده با نحو، گرامر عبارت‌های ریاضی را در نظر می‌گیریم. ترجمه‌ی هدایت شده با نحو، به هر ناپایانه یک خصیصه به نام val نسبت می‌دهد.

PRODUCTION	SEMANTIC RULE
$L \to E\mathsf{n}$	$\mathit{print}(E.val)$
$E \to E_1 + T$	$E.val := E_1.val + T.val$
$E \to T$	$E.val := T.val$
$T \to T_1 * F$	$T.val := T_1.val * F.val$
$T \to F$	$T.val := F.val$
$F \to (E)$	$F.val := E.val$
$F \to \mathsf{digit}$	$F.val :=\mathsf{digit}.\mathit{lexval}$

۳-۱۳ تعریف‌های هدایت شده با نحو

تعریف‌های هدایت شده با نحو تعمیمی از گرامرهای مستقل از متن است که در آن:

۱) نمادهای گرامر مجموعه‌ای از **خصیصه‌ها (Attributes)**ی منتسب به خود دارند.

۲) قواعد تولید گرامر برای محاسبه‌ی مقادیر خصیصه‌ها به **قواعد معنایی (Semanric rules)** نسبت داده شده‌اند.

خروجی این فرمول‌بندی، **درخت‌های تجزیه‌ی حاشیه‌نویسی شده (Annotated)** است، که در آن هر گره از نوع رکورد با یک فیلد به ازای هر خصیصه است.
(برای مثال، $X.a$ خصیصه‌ی a برای نماد گرامر X را نشان می‌دهد.)

- مقدار یک خصیصه از یک نماد گرامر در یک گره از درخت تجزیه‌ی داده شده، توسط قاعده‌ی معنایی وابسته به قاعده‌ی تولید استفاده شده در آن گره تعریف می‌شود.
- دو نوع خصیصه‌ی متمایز می‌تواند وجود داشته باشد:

۱) **خصیصه‌های ترکیبی (Synthesized Attributes)** :
که مقدار آنها از روی مقدار خصیصه‌های گره‌های فرزند آنها محاسبه می‌شود.

۲) **خصیصه‌های موروثی (Inherited Attributes)** :
که مقدار آنها از روی مقدار خصیصه‌های گره‌های همزاد و والد آنها محاسبه می‌شود.

۱-۳-۱۳ شکل‌های تعریف‌های هدایت شده با نحو

هر قاعده‌ی تولید $A \to \alpha$ به مجموعه‌ای از قواعد معنایی به شکل

$$b := f(c_1, c_2, \ldots, c_k)$$

وابسته می‌شود که در آن f یک تابع است و

۱) b یک خصیصه‌ی **ترکیبی** از A است و c_1، c_2، ...، c_k خصیصه‌های نمادهای گرامری موجود در قاعده‌ی تولید است، یا

۲) b یک خصیصه‌ی **موروثی** از یک نماد گرامر در α است و c_1، c_2، ...، c_k خصیصه‌های نمادهای گرامرهای موجود در α یا خصیصه‌های A است.

- **تذکر:** فرض می‌شود نمادهای پایانه دارای خصیصه‌هایی ترکیبی هستند که توسط تحلیل‌گر لغوی فراهم می‌شود.
- فراخوانی توابع (مانند $print$) مقادیری فرضی (Dummy) برای خصیصه‌های ترکیبی ناپایانه‌ی سمت چپ قاعده‌ی تولید، تعریف می‌کنند.

مثال

به عنوان نمونه‌ای از تعریف‌های هدایت شده با نحو، گرامر عبارت‌های ریاضی را در نظر می‌گیریم. ترجمه‌ی هدایت شده با نحو، به هر ناپایانه یک خصیصه‌ی ترکیبی به نام val نسبت می‌دهد.

PRODUCTION	SEMANTIC RULE
$L \to E$n	*print*$(E.val)$
$E \to E_1 + T$	$E.val := E_1.val + T.val$
$E \to T$	$E.val := T.val$
$T \to T_1 * F$	$T.val := T_1.val * F.val$
$T \to F$	$T.val := F.val$
$F \to (E)$	$F.val := E.val$
$F \to$ digit	$F.val :=$digit.*lexval*

۱۳-۳-۲ تعریف‌های S-خصیصه‌ای (S-Attributed)

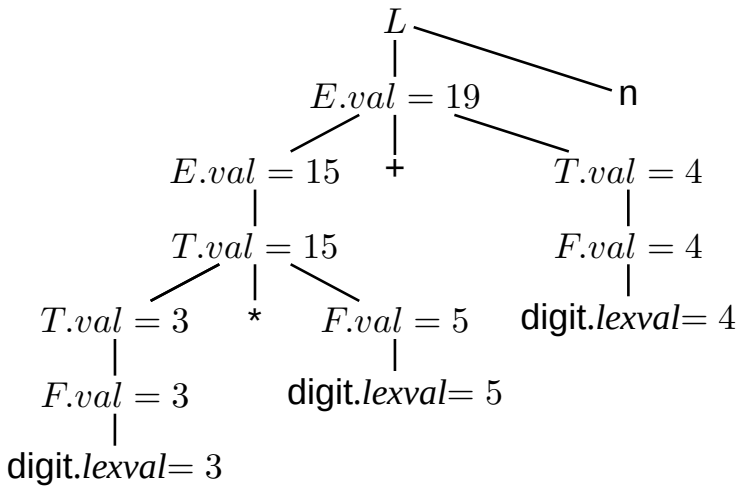
- یک **تعریف S-Attributed** یک تعریف هدایت شده با نحو است که تنها از خصیصه‌های ترکیبی استفاده می‌کند.
- **ترتیب ارزیابی**: قواعد معنایی در یک تعریف S-Attributed می‌تواند با یک پیمایش پایین به بالا یا پس‌ترتیب (Post-order) درخت تجزیه ارزیابی شود.

مثال

گرامر عبارت‌های ریاضی زیر نمونه‌ای از یک تعریف S-Attributed است.

PRODUCTION	SEMANTIC RULE
$L \to E\mathsf{n}$	$\mathit{print}(E.val)$
$E \to E_1 + T$	$E.val := E_1.val + T.val$
$E \to T$	$E.val := T.val$
$T \to T_1 * F$	$T.val := T_1.val * F.val$
$T \to F$	$T.val := F.val$
$F \to (E)$	$F.val := E.val$
$F \to \mathsf{digit}$	$F.val :=$digit.*lexval*

درخت تجزیه‌ی حاشیه‌نویسی شده برای عبارت $3 * 5 + 4\mathsf{n}$ به صورت زیر است:



۱۳-۳-۳ تعریف‌های L-خصیصه‌ای (L-Attributed)

- یک تعریف هدایت شده با نحو، **تعریف L-Attributed** نامیده می‌شود اگر هر خصیصه‌ی موروثی از X_j در قاعده‌ی تولید $A \to X_1 \dots X_j \dots X_n$ تنها وابسته باشد به:

۱) خصیصه‌های نمادهای **سمت چپ** X_j یعنی $X_1X_2\dots X_{j-1}$ و
۲) خصیصه‌های موروثی A.

- **ترتیب ارزیابی**: خصیصه‌های موروثی در تعریف‌های L-Attributed می‌تواند با یک پیمایش پیش‌ترتیب (Pre-order) و (بالا به پایین) درخت تجزیه محاسبه شوند.

خصیصه‌های موروثی

- **خصیصه‌های موروثی** برای بیان وابستگی یک ساختار به متنی که در آن ظاهر شده است، مفید می‌باشند.
- همیشه امکان‌پذیر است که یک تعریف هدایت شده با نحو به گونه‌ای بازنویسی شود که تنها خصیصه‌های ترکیبی را به کار گیرد، اما معمولاً طبیعی‌تر است که از هر دو نوع خصیصه‌ی ترکیبی و موروثی استفاده شود.
- **ترتیب ارزیابی**: خصیصه‌های موروثی نمی‌توانند توسط یک پیمایش پیش‌ترتیب ساده‌ی درخت تجزیه ارزیابی شوند:
  - بر خلاف خصیصه‌های ترکیبی، ترتیبی که با آن خصیصه‌های موروثی فرزندان محاسبه می‌گردد مهم است.
  - در واقع، خصیصه‌های موروثی فرزندان می‌تواند به همزادهای سمت چپ و راست وابسته باشد.

مثال

یک تعریف هدایت شده با نحو برای گرامر «اعلان نوع» را در نظر می‌گیریم که در آن هم خصیصه‌ی ترکیبی و هم خصیصه‌ی موروثی وجود دارد:

PRODUCTION	SEMANTIC RULE
$D \to TL$	$L.in := T.type$
$T \to$ int	$T.type :=$ *integer*
$T \to$ real	$T.type :=$ *real*
$L \to L_1$, id	$L_1.in := L.in$; *addtype(*id.*entry, L.in)*
$L \to$ id	*addtype(*id.*entry, L.in)*

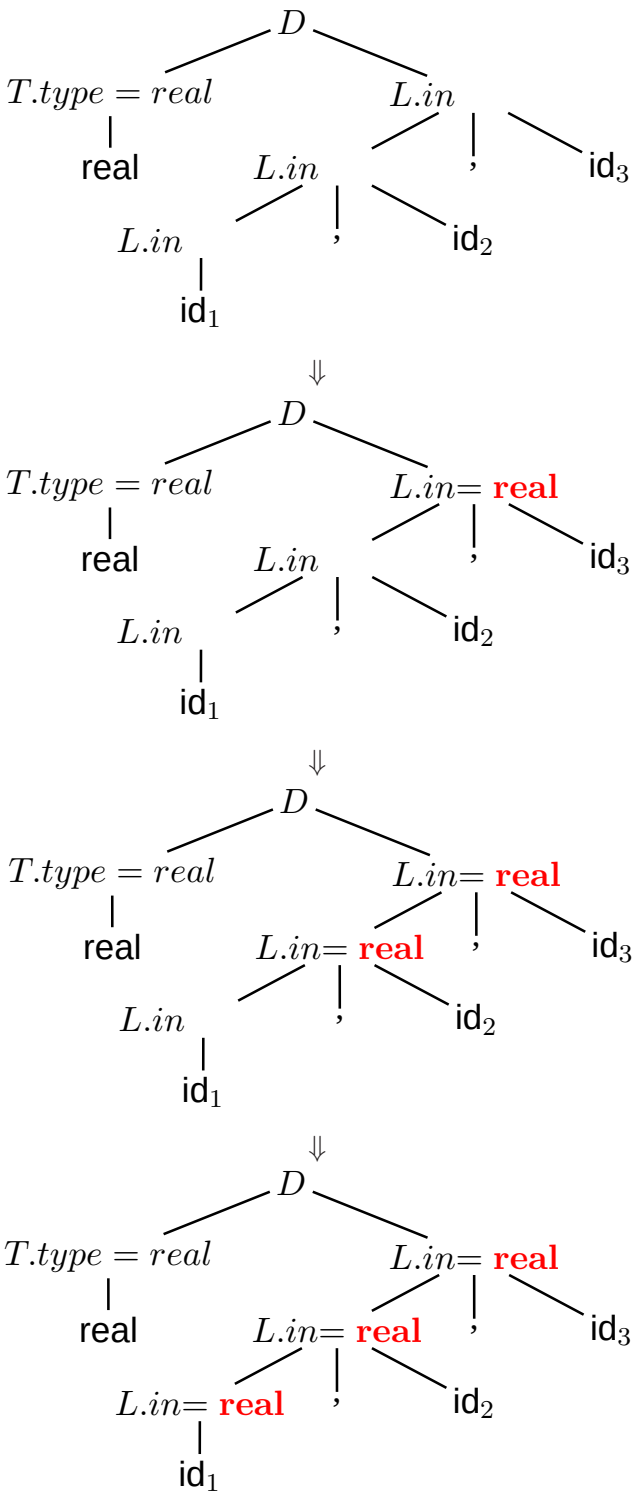
- ناپایانه‌ی T خصیصه‌ی ترکیبی $type$ را دارد که با یک کلمه‌ی کلیدی در اعلان تعیین می‌شود.
- قاعده‌ی تولید $D \to TL$ به قاعده‌ی معنایی $L.in := T.type$ وابسته شده است که خصیصه‌ی موروثی $L.in$ را مقداردهی می‌کند.
- در قاعده‌ی تولید $L \to L_1$, id دو مورد وقوع L با اندیس متمایز شده است.

- خصیصه‌های ترکیبی می‌توانند با یک پیمایش **پس‌ترتیب** ارزیابی شوند.

- خصیصه‌های موروثی که به همزادان سمت راست وابسته نیستند، می‌توانند با پیمایش عمقی ساده (پیش‌ترتیب) ارزیابی شوند.

مثال

درخت تجزیه‌ی حاشیه‌نویسی شده برای ورودی real id_1, id_2, id_3 با توجه به مثال قبل، به صورت زیر است:



- $L.in$ از بالا به پایین درخت توسط دیگر گره‌های L به ارث برده می‌شود.
- در هر گره‌ی L روال $addtype$ نوع شناسه را در جدول نمادها درج می‌کند.
- ترتیب محاسبه‌ی خصیصه‌ها با پیمایش عمقی نشان داده شده است.

۴-۳-۱۳ بازنویسی تعریف برای استفاده از خصیصه‌های ترکیبی

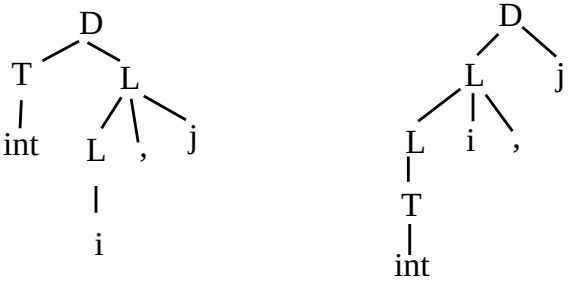
همیشه امکان‌پذیر است که یک تعریف هدایت شده با نحو به گونه‌ای بازنویسی شود که تنها خصیصه‌های ترکیبی را به کار گیرد، اما معمولاً طبیعی‌تر است که از هر دو نوع خصیصه‌ی ترکیبی و موروثی استفاده شود.

مثال

استفاده از خصیصه‌های موروثی برای دنبال کردن نوع فهرستی از متغیرها (بازنویسی گرامر):

- $D \rightarrow TL$
- $T \rightarrow int \mid char$
- $L \rightarrow L, id \mid id$

- $D \rightarrow L\ id$
- $L \rightarrow L\ id, \mid T$
- $T \rightarrow int \mid char$



۵-۳-۱۳ پیاده‌سازی ارزیابی خصیصه‌ها

گراف‌های وابستگی

برای پیاده‌سازی یک تعریف هدایت شده با نحو، ضروری است که ترتیبی برای ارزیابی خصیصه‌ها پیدا شود، زیرا مقدار هر خصیصه باید در هنگام انجام یک محاسبه بر روی آن مشخص باشد.

گراف وابستگی عمومی‌ترین تکنیکی است که برای ارزیابی تعریف‌های هدایت شده با نحو با خصیصه‌های ترکیبی و موروثی استفاده می‌شود. یک گراف وابستگی، وابستگی میان خصیصه‌های گره‌های مختلف یک درخت تجزیه را نشان می‌دهد. در این گراف

- برای هر خصیصه یک گره وجود دارد.

- اگر خصیصه‌ی b وابسته به خصیصه‌ی c باشد، یک یال از گره‌ی مربوط به c به گره‌ی مربوط به b برقرار می‌شود $(b \leftarrow c)$.

قاعده‌ی وابستگی: اگر خصیصه‌ی b وابسته به خصیصه‌ی c باشد، در این صورت ابتدا باید قاعده‌ی معنایی مربوط به c اجرا شود و سپس قاعده‌ی معنایی مربوط به b.

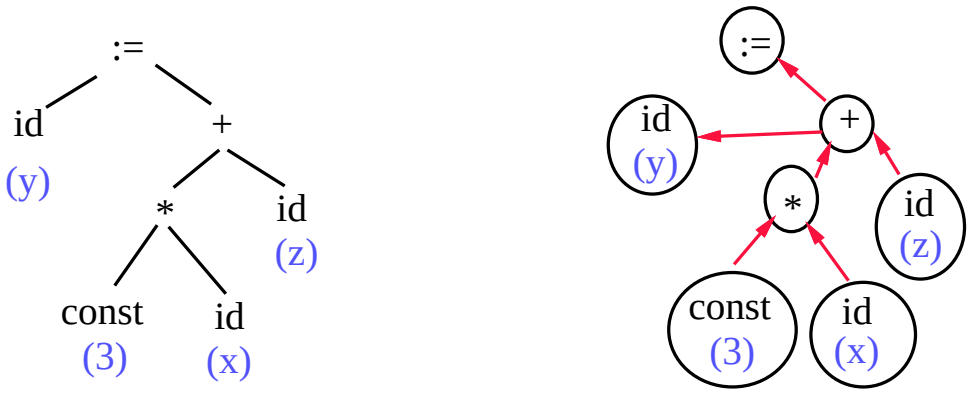
ترتیب ارزیابی

ترتیب ارزیابی قواعد معنایی، به **ترتیب توپولوژیکی** مشتق شده از گراف وابستگی، وابسته است.

- **ترتیب توپولوژیکی (Topological Sort)**:
  هر ترتیبی به صورت $m_1, m_2, \ldots, m_k$ به طوری که اگر $m_i \rightarrow m_j$ یک یال در گراف وابستگی بود، در این صورت $m_i \prec m_j$ باشد.
- هر ترتیب توپولوژیکی یک گراف وابستگی، یک ترتیب معتبر برای ارزیابی قواعد معنایی ارایه می‌دهد.

مثال

شکل زیر، یک گراف وابستگی متناظر با یک عبارت ریاضی که درخت نحو آن رسم شده است را نشان می‌دهد.



خصیصه‌ها می‌توانند با ایجاد یک گراف وابستگی و سپس یافتن یک ترتیب توپولوژیکی از آن، در زمان کامپایل ارزیابی شوند. اما این رویکرد مشکلاتی دارد:

۱) اگر گراف وابستگی دارای چرخه (cycle) باشد، این روش شکست می‌خورد؛ بنابراین به آزمونی برای تشخیص عدم وجود چرخه نیاز داریم.

۲) به دلیل ساخت گراف وابستگی، این روش زمان‌گیر است.

یک رویکرد جایگزین: طراحی تعریف هدایت شده با نحو به گونه‌ای که خصیصه‌ها بتوانند با یک ترتیب ثابت ارزیابی شوند، می‌تواند از ساخت گراف وابستگی جلوگیری کند. (این روش در اکثر کامپایلرها استفاده می‌شود.)

۱۳-۳-۶ تعریف‌های هدایت شده با نحو اکیداً بدون چرخه

تعریف هدایت شده با نحو اکیداً بدون چرخه (Strongly Non-Circular SDD) ساز و کاری است که با آن می‌توان ترتیب ارزیابی خصیصه‌ها را در زمان ساخت کامپایلر تثبیت کرد.

- این نوع تعریف‌ها، حالت‌های خاصی از تعریف‌های بدون چرخه هستند.
- دو نوع تعریف اکیداً بدون چرخه عبارت است از:
  - تعریف‌های S-Attributed
  - تعریف‌های L-Attributed

ارزیابی تعریف‌های S-خصیصه‌ای (S-Attributed)

خصیصه‌های ترکیبی می‌توانند با یک تجزیه‌گر پایین به بالا در حین تحلیل ورودی ارزیابی شوند، بدون اینکه به ساخت گراف وابستگی نیاز باشد.

- تجزیه‌گر مقدار خصیصه‌های ترکیبی را در پشته‌ی خود نگهداری می‌کند.
- هرگاه یک کاهش $A \to \alpha$ صورت می‌گیرد، خصیصه‌ی مربوط به A از روی خصیصه‌های مربوط به α که بر روی پشته ظاهر شده است، محاسبه می‌شود.
- بنابراین، یک مترجم برای یک تعریف S-Attributed می‌تواند به سادگی با گسترش پشته‌ی یک تجزیه‌گر LR پیاده‌سازی شود.

برای گسترش پشته‌ی تجزیه‌گر:

- فیلدهای اضافی به پشته اضافه می‌شود تا بتواند مقادیر خصیصه‌های ترکیبی را نگهداری کند.
- در ساده‌ترین حالت یک خصیصه به ازای هر نماد گرامر، پشته دو فیلد دارد: $state$ و val:

state	*val*
Z	$Z.x$
Y	$Y.x$
X	$X.x$
...	...

- بالای فعلی پشته، با اشاره‌گر top نشان داده می‌شود.
- خصیصه‌های ترکیبی دقیقاً پیش از هر کاهش محاسبه می‌شوند.

برای مثال: پیش از انجام کاهش $A \to XYZ$ خصیصه‌ی A محاسبه می‌شود:

$$A.a := f(val[top], val[top-۱], val[top-۲])$$

مثال

برای مشاهده‌ی چگونگی گسترش پشته‌ی تجزیه‌گر LR، تعریف‌های S-Attributed را برای عبارت‌های محاسباتی در نظر بگیرید. برای ارزیابی خصیصه‌ها، تجزیه‌گر کد زیر را اجرا می‌کند:

Production	Code
$L \to E\mathsf{n}$	$\mathit{print}(val[top-1])$
$E \to E_1 + T$	$val[ntop] := val[top] + val[top-2]$
$E \to T$	
$T \to T_1 * F$	$val[ntop] := val[top] * val[top-2]$
$T \to F$	
$F \to (E)$	$val[ntop] := val[top-1]$
$F \to \mathsf{digit}$	

- متغیر $ntop$ با «بالای جدید پشته» مقداردهی می‌شود. پس از انجام یک کاهش داریم: $top := ntop$.
- وقتی یک کاهش $A \to \alpha$ با $|\alpha| = r$ صورت می‌گیرد، خواهیم داشت: $ntop := top - r + ۱$.

فرآیند ترجمه‌ی هدایت شده با نحو برای یک رشته‌ی ورودی، به صورت زیر دنبال می‌شود:

Input	*state*	*val*	Production Used
$3 * 5 + 4\mathsf{n}$	_	_	
$*5 + 4\mathsf{n}$	3	3	
$*5 + 4\mathsf{n}$	F	3	$F \to \mathbf{digit}$
$*5 + 4\mathsf{n}$	T	3	$T \to F$
$5 + 4\mathsf{n}$	$T*$	3 _	
$+4\mathsf{n}$	$T * 5$	3 _ 5	
$+4\mathsf{n}$	$T * F$	3 _ 5	$F \to \mathbf{digit}$
$+4\mathsf{n}$	T	15	$T \to T * F$
$+4\mathsf{n}$	E	15	$E \to T$
$4\mathsf{n}$	$E+$	15 _	
n	$E + 4$	15 _ 4	
n	$E + F$	15 _ 4	$F \to \mathbf{digit}$
n	$E + T$	15 _ 4	$T \to F$
n	E	19	$E \to E + T$
	$E\mathsf{n}$	19 _	
	L	19	$L \to E\mathsf{n}$

- حالت‌های پشته با نمادهای گرامری متناظر با آنها جایگزین شده است.
- به جای توکن **digit** مقدار واقعی آن نشان داده شده است.

ارزیابی تعریف‌های L-خصیصه‌ای (L-Attributed)

تعریف‌های L-Attributed هم شامل خصیصه‌های ترکیبی و هم خصیصه‌های موروثی است اما برای ارزیابی آنها نیازی به ایجاد گراف وابستگی نیست.

- تعریف‌های L-Attributed دسته‌ای از تعریف‌های هدایت شده با نحو هستند که همیشه می‌توانند با یک بار پیمایش درخت تجزیه ارزیابی شوند.
- الگوریتم زیر، تعریف‌های L-Attributed را با ترکیب پیمایش‌های پس‌ترتیب (برای خصیصه‌های ترکیبی) و پیش‌ترتیب (برای خصیصه‌های موروثی) ارزیابی می‌کند:

Algorithm L-Eval(n: Node).
Input: Node of an annotated parse-tree.
Output: Attribute evaluation.
Begin
  For each child m of n, from left-to-right Do Begin;
    Evaluate inherited attributes of m;
    L-Eval(m)
  End;
  Evaluate synthesized attributes of n
End.

۴-۱۳ شِماهای ترجمه

شماهای ترجمه (Translation Schemes)، نسبت به تعریف‌های هدایت شده با نحو بیشتر پیاده‌سازی محور هستند، زیرا ترتیب ارزیابی قواعد معنایی و خصیصه‌ها را مشخص می‌کند.

تعریف: یک شمای ترجمه، یک گرامر مستقل از متن است که در آن

۱) خصیصه‌هایی به نمادهای گرامر نسبت داده شده است؛

۲) کنش‌های معنایی (Semantic Actions) بین آکولادهای باز و بسته {} محصور شده‌اند و در بخشی از سمت راست قواعد تولید گرامر قرار گرفته‌اند.

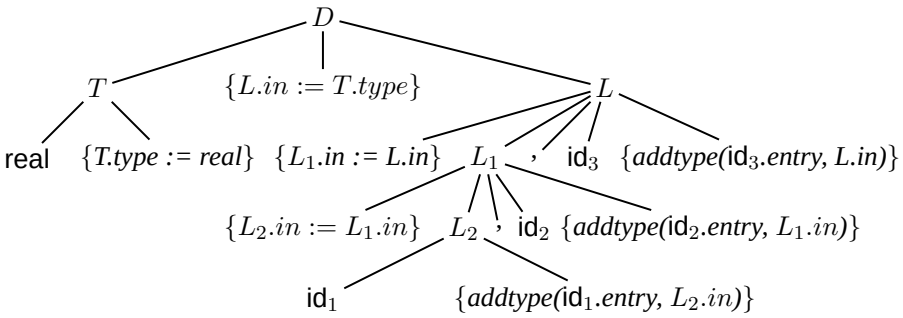
- شماهای ترجمه می‌توانند هم خصیصه‌های موروثی و هم خصیصه‌های ترکیبی داشته باشند.
- با کنش‌های معنایی مانند یک نماد پایانه برخورد می‌شود: درخت‌های تجزیه‌ی حاشیه‌نویسی شده برای هر گره شامل فرزندانی می‌باشند که حاوی کنش‌های معنایی مربوط به قاعده‌ی متناظر با آن هستند.
- شماهای ترجمه برای ارزیابی تعریف‌های L-Attributed قابل استفاده هستند (اگرچه شماهای ترجمه مکانیزم‌های عمومی‌تری هستند).

مثال

شمای ترجمه برای یک تعریف L-Attributed اعلان نوع (Type declaration) را در نظر بگیرید:

$$
\begin{aligned}
D \rightarrow &\quad T\ \{L.in := T.type\}\ L \\
T \rightarrow &\quad \text{int}\ \{T.type := integer\} \\
T \rightarrow &\quad \text{real}\ \{T.type := real\} \\
L \rightarrow &\quad \{L_1.in := L.in\}\ L_1, \text{id}\ \{addtype(\text{id}.entry, L.in)\} \\
L \rightarrow &\quad \text{id}\ \{addtype(\text{id}.entry, L.in)\}
\end{aligned}
$$

درخت تجزیه با کنش‌های معنایی برای ورودی $\text{real id}_1, \text{id}_2, \text{id}_3$ به صورت زیر است:

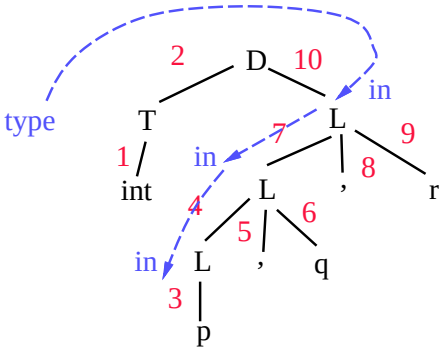


با پیمایش درخت فوق به صورت عمق ـ اول می‌توان خصیصه‌ها را ارزیابی نمود:

- $D \rightarrow T\ \{L.in := T.type\}\ L$
- $T \rightarrow int\ \{T.type := integer\}$
- $T \rightarrow real\ \{T.type := real\}$
- $L \rightarrow\ \{L_1.in := L.in\}\ L_1, id\ \{addtype(id.entry, L.in)\}$
- $L \rightarrow id\ \{addtype(id.entry, L.in)\}$

Parsing and dependency graph:

input	stack	production used
int p, q, r		
p, q, r	**int**	
p, q, r	T	$T \rightarrow int$
$, q, r$	$T\ p$	
$, q, r$	$T\ L$	$L \rightarrow id$
q, r	$T\ L$,	
$, r$	$T\ L$, q	
$, r$	$T\ L$	$L \rightarrow L, id$
r	$T\ L$,	
	$T\ L$, r	
	$T\ L$	$L \rightarrow L, id$
	D	$D \rightarrow TL$



۱۳-۴-۱ طراحی شماهای ترجمه

- وقتی می‌خواهیم یک شمای ترجمه طراحی کنیم، باید مطمئن باشیم که مقدار یک خصیصه در هنگام اجرای یک کنش آماده باشد.
- یک حالت خاص زمانی است که کنش معنایی تنها حاوی خصیصه‌های ترکیبی است: در این صورت کنش باید در انتهای قاعده‌ی تولید قرار داده شود.

مثال

قاعده‌ی تولید و قاعده‌ی معنایی زیر

$$T \to T_1 * F \qquad T.val := T_1.val * F.val$$

شمای ترجمه‌ی زیر را به دست می‌دهد:

$$T \to T_1 * F \qquad \{T.val := T_1.val * F.val\}$$

قواعد زیر برای طراحی شماهای ترجمه قابل توجه است:

- اگر هم خصیصه‌های ترکیبی و هم خصیصه‌های موروثی داشته باشیم، باید محدودیت‌های زیر را اعمال کنیم:

۱) یک خصیصه‌ی موروثی برای یک نماد واقع در سمت راست یک قاعده‌ی تولید، باید درکنشی که پیش از آن نماد قرار دارد، محاسبه شود.

۲) یک کنش نباید به یک خصیصه‌ی ترکیبی از یک نماد که در سمت راست آن کنش قرار دارد ارجاع دهد.

۳) یک خصیصه‌ی ترکیبی برای ناپایانه‌ی سمت چپ، تنها زمانی می‌تواند محاسبه شود که همه‌ی خصیصه‌هایی که بدان‌ها ارجاع می‌کند، محاسبه شده باشند. کنش مربوط معمولاً در انتهای قاعده‌ی تولید قرار داده می‌شود.

- همیشه این امکان وجود دارد که با یک تعریف L-Attributed شروع کنیم و یک شمای ترجمه ایجاد کنیم که خصوصیات فوق را ارضا می‌کند.

۱۳-۴-۲ ارزیابی شماهای ترجمه در زمان کامپایل

در یک شمای ترجمه که از قواعد فوق پیروی می‌کند، خصیصه‌ها می‌توانند در زمان کامپایل مشابه با ارزیابی تعریف‌های S-Attributed ارزیابی شوند.

ایده‌ی اصلی: از یک شمای ترجمه با کنش‌های تعبیه شده در آن آغاز می‌کنیم، سپس تبدیل‌هایی را اعمال می‌کنیم که همه‌ی کنش‌ها در انتهای سمت راست قواعد ظاهر شوند.

- یک علامت‌گذار جدید (Marker) (که در واقع یک ناپایانه مانند M است) را با کاهش به تهی برای هر کنش معنایی تعبیه شده معرفی می‌کنیم.
- کنش معنایی در انتهای قاعده‌ی تولید $M \to \epsilon$ الصاق می‌شود.

مثال

شمای ترجمه‌ی زیر را در نظر بگیرید:

$$\begin{aligned} S &\rightarrow aA\ \{C.i = f(A.s)\}\ C \\ S &\rightarrow bAB\ \{C.i = f(A.s)\}\ C \\ C &\rightarrow c\ \{C.s = g(C.i)\} \end{aligned}$$

دو علامت‌گذار $\mathbf{M_1}$ و $\mathbf{M_2}$ را با قواعد زیر اضافه می‌کنیم:

$$\begin{aligned} S &\rightarrow aA\mathbf{M_1}C \\ S &\rightarrow bAB\mathbf{M_2}C \\ \mathbf{M_1} &\rightarrow \epsilon \quad \{\mathbf{M_1}.s := f(val[top])\} \\ \mathbf{M_2} &\rightarrow \epsilon \quad \{\mathbf{M_2}.s := f(val[top-1])\} \\ C &\rightarrow c \quad \{C.s := g(val[top-1])\} \end{aligned}$$

- ملاحظه می‌شود که خصیصه‌ی موروثی $C.i$ به خصیصه‌ی ترکیبی $\mathbf{M_1}.s$ یا $\mathbf{M_2}.s$ نگاشت داده شده است.
- با توجه به قواعد، هرگاه $C \rightarrow c$ استفاده می‌شود، مقدار $C.i$ در $val[top-1]$ قرار دارد (زیرا $C.i$ اکنون در $\mathbf{M_1}.s$ یا $\mathbf{M_2}.s$ واقع است و محتوای پشته پس از استفاده از قاعده‌ی $C \rightarrow c$ یکی از دو حالت $aA\mathbf{M_1}c$ یا $bAB\mathbf{M_2}c$ را دارد و در هر دو علامت‌گذار دومین عنصر بالای پشته یعنی در اندیس $top-1$ است).
- همچنین وقتی قاعده‌ی $\mathbf{M_1} \rightarrow \epsilon$ استفاده می‌شود، حالت پشته به صورت $aA\mathbf{M_1}$ است، در نتیجه مقدار خصیصه‌ی ترکیبی $\mathbf{M_1}.s$ که همان $C.i$ است و تابعی از $A.s$ است، از روی $val[top]$ محاسبه می‌شود، زیرا A نماد بالای پشته بلافاصله پس از استفاده از قاعده‌ی ϵ است (توجه کنید کنش معنایی دقیقاً پس از استفاده از قاعده اجرا می‌شود).
- به طور مشابه، وقتی قاعده‌ی $\mathbf{M_2} \rightarrow \epsilon$ استفاده می‌شود، حالت پشته به صورت $bAB\mathbf{M_2}$ است، در نتیجه مقدار خصیصه‌ی ترکیبی $\mathbf{M_2}.s$ که همان $C.i$ است و تابعی از $A.s$ است، از روی $val[top-1]$ محاسبه می‌شود، زیرا A دومین نماد بالای پشته بلافاصله پس از استفاده از قاعده‌ی ϵ است (دوباره توجه کنید کنش معنایی دقیقاً پس از استفاده از قاعده اجرا می‌شود).

قواعد عمومی. برای ارزیابی ترجمه‌ها در حین تجزیه‌ی پایین به بالا با فرض یک گرامر L-Attributed:

- برای هر قاعده‌ی تولید $A \rightarrow X_1 \dots X_n$، n علامت‌گذار جدید $M_1, \dots, M_n$ معرفی می‌کنیم و آن قاعده را با قاعده‌ی $A \rightarrow M_1X_1 \dots M_nX_n$ جایگزین می‌شود.
- بنابراین، مکان هر خصیصه‌ی ترکیبی و موروثی X_j و A را می‌دانیم:
  ۱) $X_j.s$ در مدخل val در پشته‌ی تجزیه‌گر وابسته به X_j ذخیره می‌شود.
  ۲) $X_j.i$ در مدخل val در پشته‌ی تجزیه‌گر وابسته به M_j ذخیره می‌شود.
  ۳) $A.i$ در مدخل val بلافاصله پیش از محل ذخیره‌سازی در پشته‌ی تجزیه‌گر ذخیره می‌شود.
- **ملاحظه ۱:** هر گرامر $\mathrm{LL}(1)$ با علامت‌گذار، همچنین یک گرامر $\mathrm{LR}(1)$ هم خواهد بود.
- **ملاحظه ۲:** یک گرامر $\mathrm{LR}(1)$ با علامت‌گذار، می‌تواند حاوی تداخل باشد!!

مثال

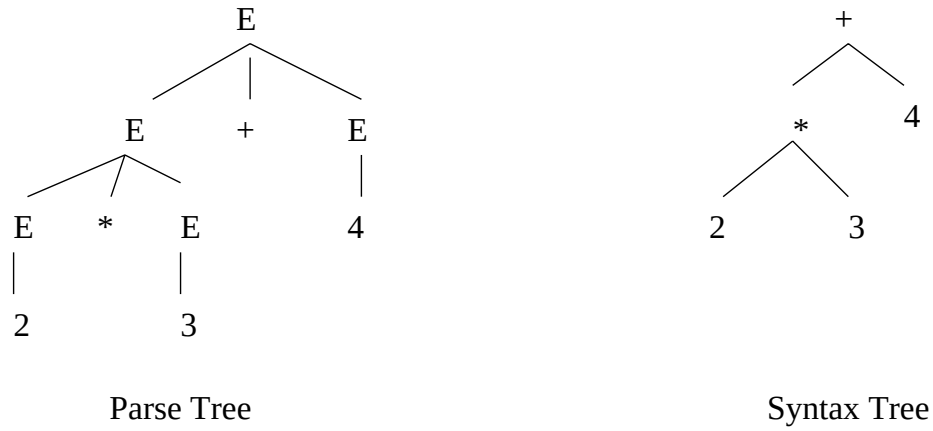
Example. Computing the inherited attribute $X_j.i$ after reducing with $M_j \to \epsilon$.

	M_j	$X_j.i$
top→	X_{j-1}	$X_{j-1}.s$
	M_{j-1}	$X_{j-1}.i$
	...	...
	X_1	$X_1.s$
	M_1	$X_1.i$
(*top-2j+2*)→	M_A	$A.i$
(*top-2j*)→		

- $A.i$ is in $val[top - 2j + 2]$;
- $X_1.i$ is in $val[top - 2j + 3]$;
- $X_1.s$ is in $val[top - 2j + 4]$;
- $X_2.i$ is in $val[top - 2j + 5]$;
- and so on.

درخت نحو

درخت نحو (Syntax Tree) یک نمایش فشرده از درخت تجزیه (Parse Tree) است که در آن ناپایانه‌ها کنار گذاشته شده‌اند.



شکل ۱۳-۱: نمونه‌ای از درخت تجزیه (چپ) و درخت نحو معادل با آن (راست)

◄ تمرین

۱. با استفاده از تعریف هدایت شده با نحو زیر، درخت تجزیه‌ی حاشیه‌نویسی شده را برای رشته‌های زیر به دست آورید:

آ) $(3+4)*(5+6)\mathsf{n}$

ب) $1*2*3*(4+5)\mathsf{n}$

PRODUCTION	SEMANTIC RULE
$L \to E\mathsf{n}$	*print*$(E.val)$
$E \to E_1 + T$	$E.val := E_1.val + T.val$
$E \to T$	$E.val := T.val$
$T \to T_1 * F$	$T.val := T_1.val * F.val$
$T \to F$	$T.val := F.val$
$F \to (E)$	$F.val := E.val$
$F \to$ digit	$F.val :=$digit.*lexval*

۲. با استفاده از تعریف هدایت شده با نحو زیر، درخت تجزیه‌ی حاشیه‌نویسی شده را برای رشته‌ی **real id**, **id**, **id**, **id** به دست آورید:

PRODUCTION	SEMANTIC RULE
$D \to TL$	$L.in := T.type$
$T \to$int	$T.type :=$*integer*
$T \to$real	$T.type :=$*real*
$L \to L_1,$ id	$L_1.in := L.in;$ *addtype(*id.*entry, L.in)*
$L \to$ id	*addtype(*id.*entry, L.in)*

۳. فرض می‌کنیم که قاعده‌ی تولید $A \to BCD$ را داریم. هر یک از چهار ناپایانه‌ی A، B، C و D دارای دو خصیصه هستند: s که یک خصیصه‌ی ترکیبی و i که یک خصیصه‌ی موروثی است. برای هر یک از مجموعه قواعد زیر تعیین کنید: (آ) آیا آن قواعد با یک تعریف S-attributed سازگار هستند؟ (ب) آیا آن قواعد با یک تعریف L-attributed سازگار هستند؟

آ) $A.s := B.i + C.s$

ب) $A.s := B.i + C.s;\ D.i := A.i + B.s$

پ) $A.s := B.s + D.s$

۴. گرامر زیر اعداد دودویی شامل بخش صحیح و کسری (که با یک نقطه جدا می‌شوند) را تولید می‌کند:

$$\begin{aligned} S &\to L.L \mid L \\ L &\to LB \mid B \\ B &\to 0 \mid 1 \end{aligned}$$

یک تعریف هدایت شده با نحو L-attributed طراحی کنید که $S.val$ مقدار دهدهی رشته‌ی ورودی را محاسبه می‌کند. برای مثال، ترجمه‌ی رشته‌ی 101.101 باید عدد دهدهی 5.625 شود. راهنمایی: از خصیصه‌ی موروثی $L.side$ استفاده کنید تا مشخص کند بیت مورد نظر در کدام سمت نقطه قرار دارد.

۵. ترجمه‌ی هدایت شده با نحو زیر را بازنویسی کنید به گونه‌ای که گرامر حاصل بازگشتی از چپ نداشته باشد. در اینجا a، b، c و d کنش‌ها و 0 و 1 پایانه هستند.

$$\begin{aligned} A &\rightarrow A\ \{a\}\ B \mid AB\ \{b\} \mid 0 \\ B &\rightarrow B\ \{c\}\ A \mid BA\ \{d\} \mid 1 \end{aligned}$$

۶. ترجمه‌ی هدایت شده با نحو (شمای ترجمه) زیر، مقدار رشته‌ای از صفرها و یک‌ها را به عنوان یک عدد صحیح دودویی مثبت محاسبه می‌کند:

$$\begin{aligned} B &\rightarrow B_1 0\ \{B.val := 2 \times B_1.val\} \\ &\mid B_1 1\ \{B.val := 2 \times B_1.val + 1\} \\ &\mid 1\ \{B.val := 1\} \end{aligned}$$

این تعریف را به گونه‌ای بازنویسی کنید که بازگشتی از چپ نباشد اما مقدار مشابهی برای $B.val$ برای کل رشته‌ی ورودی محاسبه نماید.

۷. یک شمای ترجمه‌ی هدایت شده با نحو برای تبدیل عبارت‌های ریاضی میانوندی با عملگرهای $+$ و $*$ و عملوندهای متشکل از ارقام 0 تا 9 به عبارت‌های ریاضی پسوندی طراحی کنید.

۸. اختیاری. گرامر زیر را برای اعداد در نظر بگیرید که در آن اعداد می‌توانند در مبنای هشت (octal نمایش داده شده با پسوند **o**) یا در مبنای ده (decimal نمایش داده شده با پسوند **d**) باشند.

$$\begin{aligned} \textit{based-num} &\rightarrow \textit{num base-char} \\ \textit{base-char} &\rightarrow \mathbf{o} \\ \textit{base-char} &\rightarrow \mathbf{d} \\ \textit{num} &\rightarrow \textit{num digit} \\ \textit{num} &\rightarrow \textit{digit} \\ \textit{digit} &\rightarrow \mathbf{0} \mid \mathbf{1} \mid \ldots \mid \mathbf{9} \end{aligned}$$

آ) یک گرامر خصیصه‌دار (تعریف هدایت شده با نحو) برای تعیین مقدار عدد ارایه دهید.

ب) یک درخت تجزیه‌ی حاشیه‌نویسی شده برای **123o** رسم کنید.

پ) آیا خصیصه‌های طراحی شده توسط شما می‌توانند در حین تجزیه با یک تجزیه‌گر بالا به پایین محاسبه شوند؟

ت) یک شبه کد برای محاسبه‌ی خصیصه‌ها به صورت single-pass (با یک بار پیمایش درخت تجزیه) ارایه دهید.