اصول طراحی کامپایلر

حسین کارشناس

دانشکده مهندسی کامپیوتر

ترم اول ۹۸ – ۹۷

ترجمه دستور گرا
(syntax directed translation)

ترجمه دستور گرا

- استفاده از رویه‌های معنایی (semantic procedures) در حین تجزیه
  - برای انجام عملیات تکمیلی
    - انجام سایر مراحل بخش پیشین (frontend) کامپایلر
  - فراخوانی رویه‌های معنایی در حین ساخت درخت تجزیه
    - رویه‌ها در گره‌های بخصوصی از درخت تجزیه اجرا می‌شوند

- دو راهکار کلی
  - تعریف‌ها (syntax-directed definition – SDD)
  - طرح‌های ترجمه (syntax-directed translation scheme – SDT)

ترجمه دستور گرا

- تعریف دستور گرا (SDD)
- یک گرامر مستقل از متن به همراه مجموعه‌ای از ویژگی‌ها و قوانین معنایی
  - اختصاص ویژگی‌ها (attributes) به نشانه‌های گرامر
    - دربردارنده اطلاعاتی در مورد آن نشانه گرامر
  - اختصاص قوانین معنایی (semantic rules) به قواعد تولید گرامر
    - دنباله‌ای از دستورات اجرایی (یک تکه برنامه)
    - برای محاسبه مقادیر ویژگی‌ها مورد استفاده قرار می‌گیرند
- بکارگیری یک قاعده تولید در روند تجزیه منجر به محاسبه مقادیر ویژگی‌های مرتبط با نشانه‌های آن قاعده توسط قوانین معنایی مربوطه می‌شود

ترجمه دستور گرا

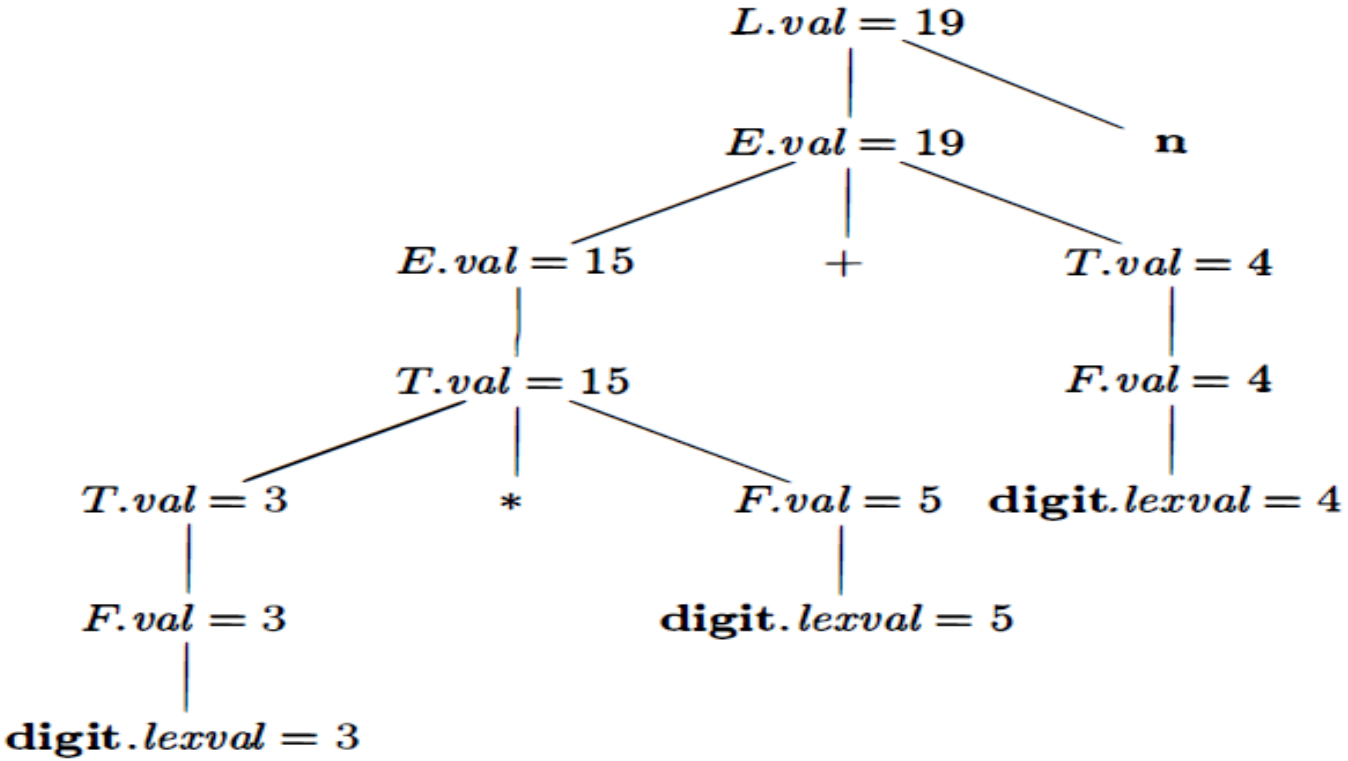
- مثال: یک SDD برای محاسبه مقدار عبارات ریاضی

	PRODUCTION	SEMANTIC RULES
1)	$L \rightarrow E$ **n**	$L.val = E.val$
2)	$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3)	$E \rightarrow T$	$E.val = T.val$
4)	$T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5)	$T \rightarrow F$	$T.val = F.val$
6)	$F \rightarrow (E)$	$F.val = E.val$
7)	$F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

- ویژگی‌ها
  - L.val، E.val، T.val، F.val، **digit**.lexval

ترجمه دستور گرا

- درخت تجزیه مشروح (annotated)
- حاوی مقادیر ویژگی‌ها در گره‌های درخت تجزیه برای یک برنامه ورودی
- مثال: درخت تجزیه مشروح برای ورودی $3 * 5 + 4\,\mathbf{n}$



ترجمه دستور گرا

- انواع ویژگی‌ها برای نشانه‌های غیرپایانی در یک گره از درخت تجزیه
  - ویژگی‌های ساختی (synthesized) در یک گره
    - مرتبط با نشانه غیرپایانی سمت چپ قاعده تولید
    - وابسته به ویژگی‌های موجود در آن گره و گره‌های فرزندان
  - ویژگی‌های موروثی (inherited) در یک گره
    - مرتبط با نشانه‌های غیرپایانی در بدنه قاعده تولید
    - وابسته به ویژگی‌های موجود در آن گره، گره پدر و گره‌های برادر
- نشانه‌های پایانی فقط دارای ویژگی‌های ساختی هستند
  - مقدار این ویژگی‌ها توسط تحلیل‌گر واژه‌ای داده می‌شود (قانون معنایی ندارد)
    - همان ویژگی‌های نماد مرتبط با نشانه پایانی

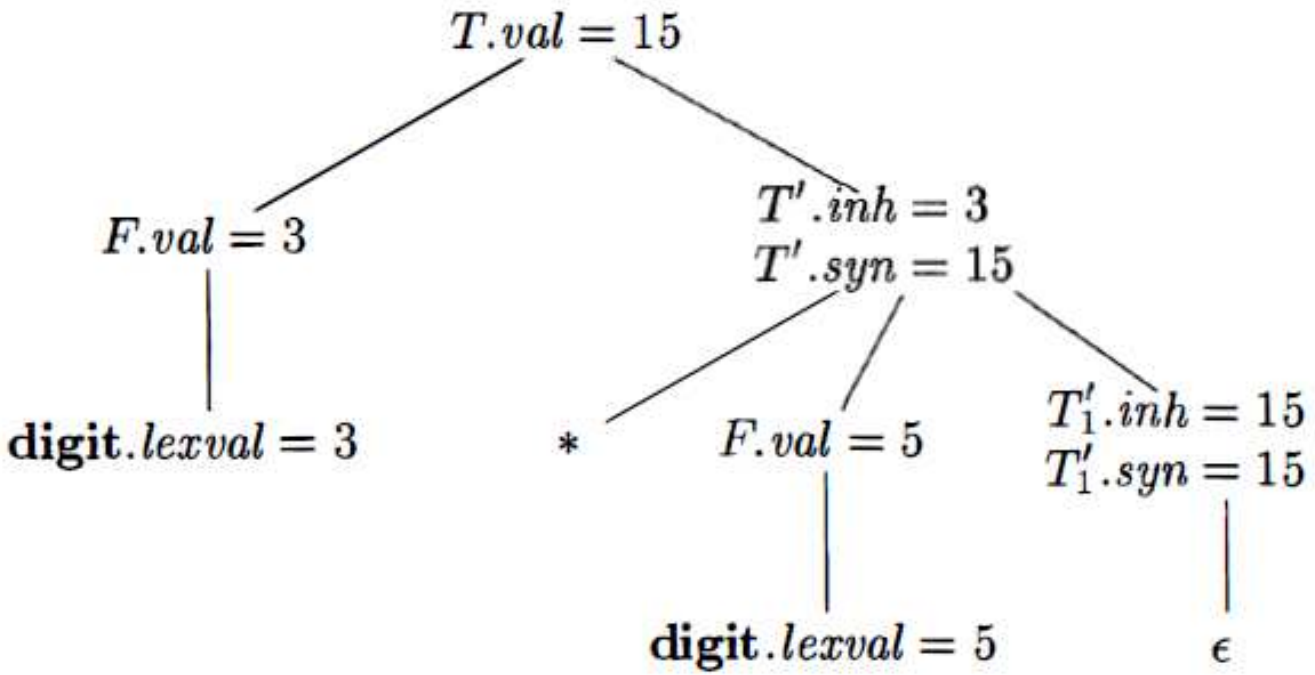
ترجمه دستور گرا

- انتقال مقادیر به شاخه‌های دیگر درخت تجزیه با ویژگی‌های موروثی
  - مثال: یک SDD دیگر برای محاسبه مقدار عبارات ریاضی
    - متناظر با یک گرامر غیربازگشتی چپ (برای تجزیه بالا به پایین)

	PRODUCTION	SEMANTIC RULES
1)	$T \rightarrow F\, T'$	$T'.inh = F.val$ $T.val = T'.syn$
2)	$T' \rightarrow * F\, T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3)	$T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4)	$F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit}.lexval$

ترجمه دستور گرا

- مثال: درخت تجزیه مشروح 5*3 برای SDD مثال قبل



- محاسبه مقدار ویژگی T´.inh با توجه به مقدار ویژگی F.val در گره برادر

طرح‌های ترجمه دستور گرا

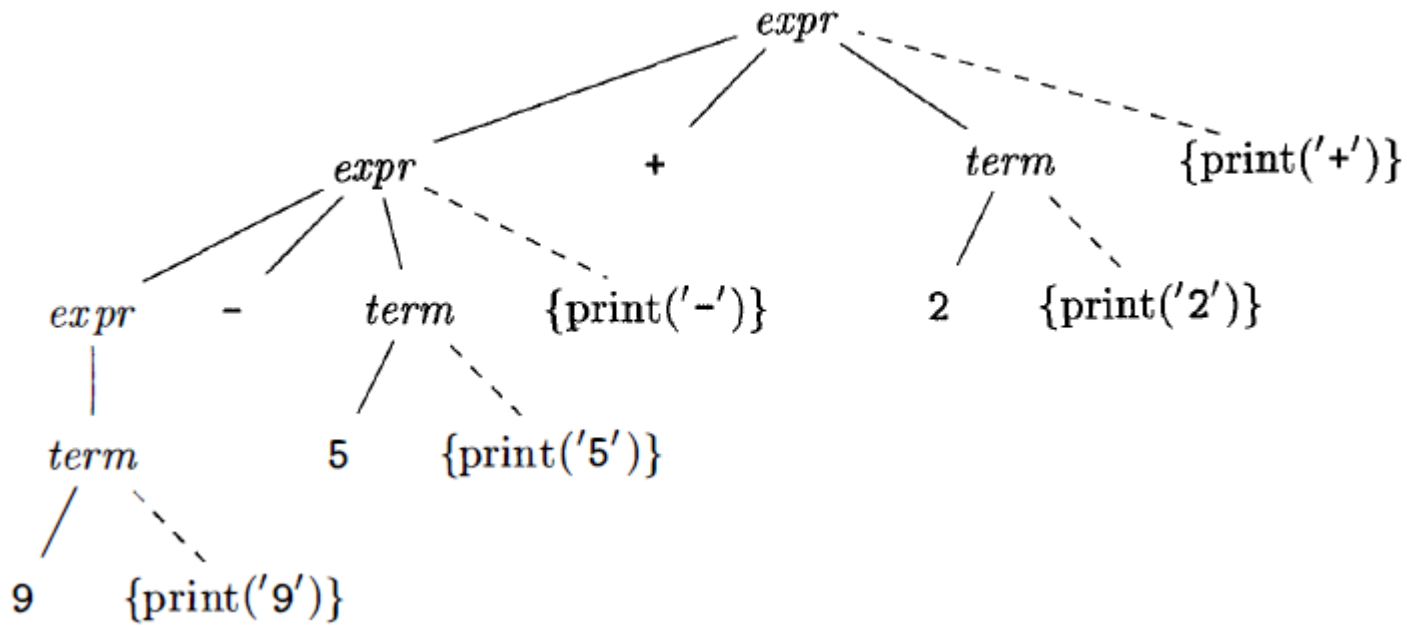
طرح‌های ترجمه

- یک گرامر مستقل از متن به همراه کنش‌های معنایی
  - هر کنش معنایی (semantic action) شامل مجموعه‌ای از دستورات
    - یک تکه برنامه
    - می‌تواند در هر جایی از بدنه قواعد تولید گرامر بیاید (مثل یک نشانه از گرامر)
- هر گونه ترجمه با SDDها بوسیله طرح‌های ترجمه هم قابل پیاده‌سازی است
- مثال: یک طرح ترجمه

$expr$	$\rightarrow$	$expr_1$ + $term$	{print('+')}
$expr$	$\rightarrow$	$expr_1$ - $term$	{print('-')}
$expr$	$\rightarrow$	$term$	
$term$	$\rightarrow$	0	{print('0')}
$term$	$\rightarrow$	1	{print('1')}
		...	
$term$	$\rightarrow$	9	{print('9')}

طرح‌های ترجمه

- مثال: درخت تجزیه حاوی کنش‌های معنایی



- ترتیب اجرای کنش‌های معنایی
  - کنش‌های معنایی باید به صورت عمق اول و با ترتیب چپ به راست اجرا شوند

طرح‌های ترجمه

- SDTها در حالت کلی
- کنش‌های معنایی می‌توانند در هر جایی از بدنه قواعد تولید قرار بگیرند
  - قواعد تولید به صورت $A \rightarrow \alpha X\{t\}Y\beta$
- کنش معنایی به محض پردازش تمام نشانه‌های سمت چپ آن اجرا می‌شود
  - در تجزیه بالا به پایین کنش t بلافاصله پیش از بسط یا تطبیق نشانه Y اجرا می‌شود
  - در تجزیه پایین به بالا کنش t به محض ظاهر شدن حالت متناظر با نشانه X در بالای پشته اجرا می‌شود
- برخی از انواع SDTها در حین تجزیه قابل پیاده‌سازی نیستند

طرح‌های ترجمه

- امکان محاسبه مقادیر ویژگی‌ها در SDTها
- کنش‌های معنایی ویژگی‌های مرتبط با نشانه‌های غیرپایانی را محاسبه می‌کنند
- مثال:

	PRODUCTION	SEMANTIC RULES
1)	$L \to E\ \mathbf{n}$	$L.val = E.val$
2)	$E \to E_1 + T$	$E.val = E_1.val + T.val$
3)	$E \to T$	$E.val = T.val$
4)	$T \to T_1 * F$	$T.val = T_1.val \times F.val$
5)	$T \to F$	$T.val = F.val$
6)	$F \to (E)$	$F.val = E.val$
7)	$F \to \mathbf{digit}$	$F.val = \mathbf{digit}.\text{lexval}$

L	$\to$	$E\ \mathbf{n}$	$\{\ \text{print}(E.val);\ \}$
E	$\to$	$E_1 + T$	$\{\ E.val = E_1.val + T.val;\ \}$
E	$\to$	T	$\{\ E.val = T.val;\ \}$
T	$\to$	$T_1 * F$	$\{\ T.val = T_1.val \times F.val;\ \}$
T	$\to$	F	$\{\ T.val = F.val;\ \}$
F	$\to$	$(\ E\ )$	$\{\ F.val = E.val;\ \}$
F	$\to$	$\mathbf{digit}$	$\{\ F.val = \mathbf{digit}.lexval;\ \}$

- اهمیت رعایت ترتیب در محاسبه مقادیر ویژگی‌ها
- ترتیب اجرای کنش‌های معنایی (محل قرارگیری آنها در بدنه قواعد تولید) باید با ترتیب صحیح محاسبه مقادیر ویژگی‌ها تطبیق داشته باشد

طرح‌های ترجمه

- SDT معادل برای SDD با ویژگی L

$$A \rightarrow X_1 \ldots X_n$$

- کنش‌های معنایی برای محاسبه ویژگی‌های موروثی یک نشانه غیرپایانی (X_i) بلافاصله قبل از آن نشانه در بدنه یک قاعده تولید قرار می‌گیرند
  - ترتیب محاسبه مقدار چند ویژگی موروثی وابسته باید قابل پیاده‌سازی باشد
  - ویژگی‌های مورد استفاده در سایر ویژگی‌ها باید ابتدا محاسبه شوند

- کنش‌های معنایی برای محاسبه ویژگی‌های ساختی نشانه سمت چپ قاعده تولید (A) در انتهای بدنه آن قاعده تولید قرار می‌گیرند

طرح‌های ترجمه

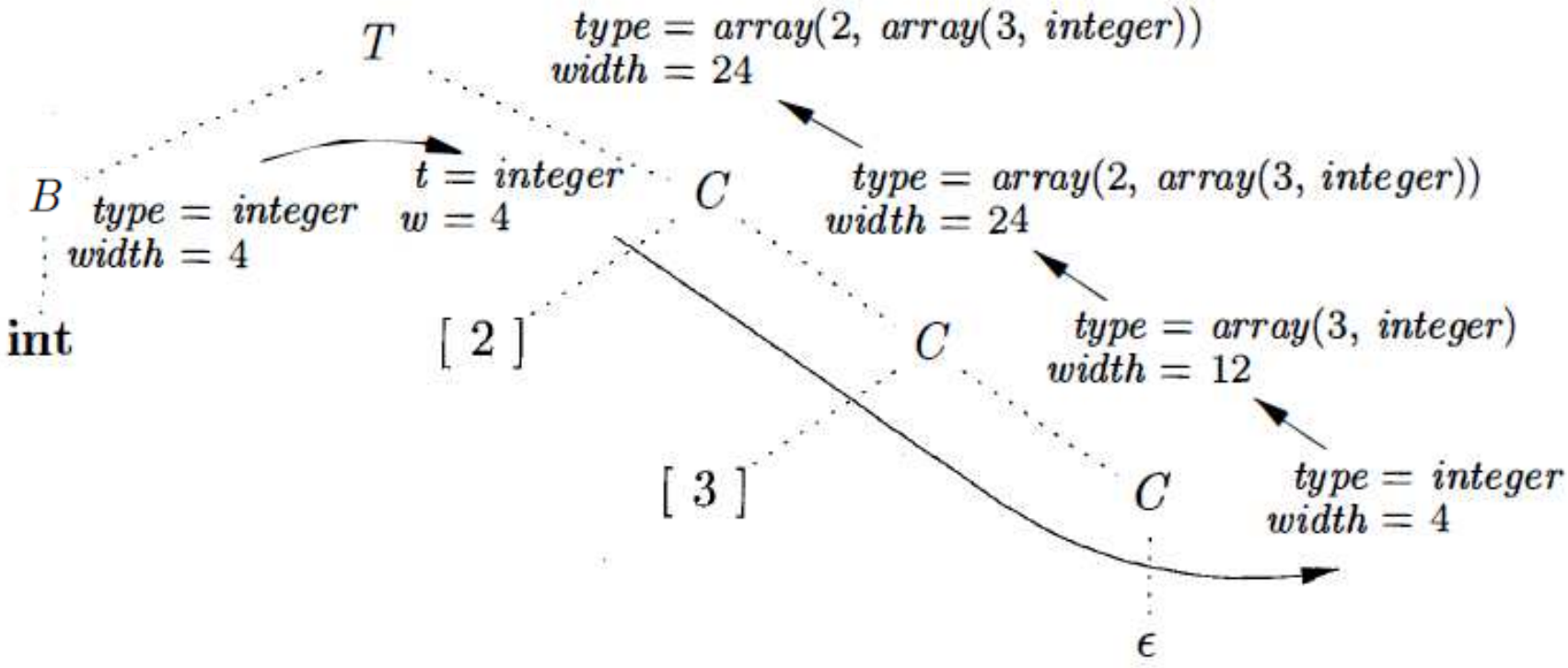
- مثال: یک SDT برای تعیین انواع و عرض هر یک از آنها

$$
\begin{array}{lll}
T \rightarrow B & & \{\ t = B.type;\ w = B.width;\ \} \\
\quad\ C & & \{\ T.Type = C.type;\ T.width = C.width;\ \} \\
B \rightarrow \textbf{int} & & \{\ B.type = integer;\ B.width = 4;\ \} \\
B \rightarrow \textbf{float} & & \{\ B.type = float;\ B.width = 8;\ \} \\
C \rightarrow \epsilon & & \{\ C.type = t;\ C.width = w;\ \} \\
C \rightarrow [\ \textbf{num}\ ]\ C_1 & & \{\ C.Type = array(\textbf{num}.value,\ C_1.type); \\
 & & \quad C.width = \textbf{num}.value \times C_1.width;\ \}
\end{array}
$$

- متغیرهای t و w مانند ویژگی‌های موروثی C مورد استفاده قرار می‌گیرند

طرح‌های ترجمه

- مثال: تعیین نوع (محاسبه عبارت نوع) و عرض برای تعریف int[2][3]



طرح‌های ترجمه

- مثال: یک SDT برای تعیین آدرس نسبی (offset) نام‌ها

$$
\begin{array}{lll}
P \rightarrow & & \{\ offset = 0;\ \} \\
 & D & \\
D \rightarrow & T\ \mathbf{id}\ ; & \{\ top.put(\mathbf{id}.lexeme,\ T.type,\ offset); \\
 & & \quad offset = offset + T.width;\ \} \\
 & D_1 & \\
D \rightarrow & \epsilon &
\end{array}
$$

- متغیر offset در ابتدای هر حوزه مقداردهی اولیه (0) می‌شود
- متغیر top نشان دهنده جدول نشانه‌های (حوزه) فعلی
  - متد put نام‌های جدید را به عنوان یک سطر جدید به جدول نشانه‌ها اضافه می‌کند

طرح‌های ترجمه

- مثال: یک SDT برای پشتیبانی از حوزه‌های متفاوت (غیر تودرتو)

$$
\begin{array}{lcll}
program & \rightarrow & & \{\ top = \textbf{null};\ \} \\
 & & block & \\
block & \rightarrow & '\{' & \{\ saved = top; \\
 & & & \quad top = \textbf{new}\ Env(top); \\
 & & & \quad \text{print}(\text{"\{ "});\ \} \\
 & & decls\ stmts\ '\}' & \{\ top = saved; \\
 & & & \quad \text{print}(\text{"\} "});\ \} \\
decls & \rightarrow & decls\ decl & \\
 & | & \epsilon & \\
decl & \rightarrow & \textbf{type id ;} & \{\ s = \textbf{new}\ Symbol; \\
 & & & \quad s.type = \textbf{type}.lexeme \\
 & & & \quad top.put(\textbf{id}.lexeme,\ s);\ \}
\end{array}
$$

جمع‌بندی

- رویکردهای پیاده‌سازی تحلیل معنایی
  - پس از تجزیه
    - با پویش درخت تجزیه
  - در حین تجزیه
    - ترجمه دستورگرا
    - بکارگیری رویه‌های معنایی (قواعد یا کنش‌های معنایی) در گرامر
    - اهمیت ترتیب صحیح اجرای رویه‌های معنایی
- اعمال قوانین معنایی زبان در تحلیل معنایی
  - نیاز به اطلاعات نشانه‌های تعریف شده در حوزه‌های مختلف برنامه ورودی