

Теория автоматов и формальных языков

Синтаксически управляемая трансляция

Автор: Екатерина Вербицкая

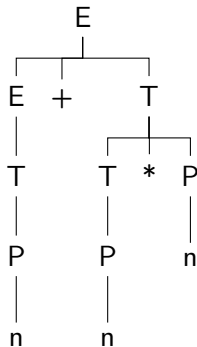
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

4 декабря 2020

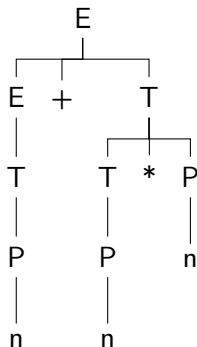
В предыдущей серии

- Что такое язык; когда предложение принадлежит языку
- Классы языков
 - ▶ Регулярные
 - ▶ Контекстно-свободные
 - ★ $LL(k)$
 - ★ $LR(k)$, $LALR(k)$
 - ▶ Задаваемые PEG
- Как задать язык
 - ▶ Конечный автомат
 - ▶ Магазинный автомат
 - ▶ PEG
- Синтаксический анализ
 - ▶ Определение, принадлежит ли цепочка языку
 - ▶ Построение дерева разбора

Дерево разбора — лишь цепочка в некотором языке



Дерево разбора — лишь цепочка в некотором языке



$[.E[.E[.T[.P[.n]]]]][.+][.T[.T[.P[.n]]][.*][.P[.n]]]$

Трансляция (перевод)

- **Трансляция** — преобразование некоторой входной строки в некоторую выходную
 - ▶ Σ — входной алфавит, Π — выходной алфавит.
Трансляцией с языка $L_i \subseteq \Sigma^*$ на язык $L_o \subseteq \Pi^*$ называется отображение $\tau : L_i \rightarrow L_o$
- Построение дерева разбора — простейший пример трансляции
- Другие примеры трансляции
 - ▶ Вычисление значения арифметического выражения
 - ▶ Преобразование арифметического выражения из инфиксной записи в постфиксную
 - ▶ Преобразование программы на языке Java в байт-код
 - ▶ Компиляция программ
 - ▶ ...
- Фактически синтаксический анализ нужен для трансляции

Схемы синтаксически управляемой трансляции

Схема синтаксически управляемой трансляции — пятерка
 (N, Σ, Π, P, S)

- N — конечное множество нетерминальных символов
- Σ — конечный входной алфавит
- Π — конечный выходной алфавит
- $S \in N$ — стартовый нетерминал
- P — конечное множество правил трансляции вида $A \rightarrow \alpha, \beta$, где $\alpha \in (N \cup \Sigma)^*, \beta \in (N \cup \Pi)^*$
 - ▶ Вхождения нетерминалов в цепочку β образуют перестановку нетерминалов из цепочки α
 - ▶ Если нетерминалы повторяются больше одного раза, то их различают по индексам: $E \rightarrow E^l + E^r, E^r + E^l$

Выводимая пара в СУ-схеме

- Если $A \rightarrow (\alpha, \beta) \in P$, то $(\gamma A^i \delta, \gamma' A^i \delta') \Rightarrow (\gamma \alpha \delta, \gamma' \beta \delta')$
- Рефлексивно-транзитивное замыкание отношения \Rightarrow называется отношением выводимости в СУ-схеме, обозначается \Rightarrow^*
- Трансляцией назовем множество пар $\{(\alpha, \beta) \mid (S, S) \Rightarrow^* (\alpha, \beta), \alpha \in \Sigma^*, \beta \in \Pi^*\}$
- СУ-схема называется простой, если во всех правилах $A \rightarrow (\alpha, \beta)$, нетерминалы в α и β встречаются в одном и том же порядке

Пример СУ-схемы

$$\begin{array}{lcl} E & \rightarrow & E + T \quad , \quad ET + \\ & | & T \quad , \quad T \\ T & \rightarrow & T * F \quad , \quad TF * \\ & | & F \quad , \quad F \\ F & \rightarrow & n \quad , \quad n \\ & | & (E) \quad , \quad E \end{array}$$

Пример СУ-схемы

$$\begin{array}{lcl} E & \rightarrow & E + T \quad , \quad ET + \\ & | & T \quad , \quad T \\ T & \rightarrow & T * F \quad , \quad TF * \\ & | & F \quad , \quad F \\ F & \rightarrow & n \quad , \quad n \\ & | & (E) \quad , \quad E \end{array}$$

$$\begin{aligned} (\underline{E}, \underline{E}) &\Rightarrow (\underline{T}, \underline{T}) \Rightarrow (\underline{T} * F, \underline{T} F *) \Rightarrow (\underline{F} * F, \underline{F} F *) \Rightarrow (n * \underline{F}, n \underline{F} *) \Rightarrow \\ (n * (\underline{E}), n \underline{E} *) &\Rightarrow (n * (\underline{E} + T), n \underline{E} T + *) \Rightarrow (n * (\underline{T} + T), n \underline{T} T + *) \Rightarrow \\ (n * (\underline{F} + T), n \underline{F} T + *) &\Rightarrow (n * (n + \underline{T}), n n \underline{T} + *) \Rightarrow \\ (n * (n + \underline{F}), n n \underline{F} + *) &\Rightarrow (n * (n + n), n n n + *) \end{aligned}$$

Обобщенные схемы синтаксически управляемой трансляции

Обобщенная схема синтаксически управляемой трансляции — шестерка $(N, \Sigma, \Pi, \Gamma, P, S)$

- Γ — конечное множество символов перевода вида $A_i, A \in N; i \in \mathbb{Z}$
- P — конечное множество правил трансляции вида $A \rightarrow \alpha, A_1 = \beta_1, \dots, A_n = \beta_n$, где $\alpha \in (N \cup \Sigma)^*$
 - ▶ $A_i \in \Gamma, 1 \leq i \leq n$
 - ▶ Каждый символ x , входящий в β_i , либо $x \in \Pi$, либо $x = B_k \in \Gamma$, где $B \in \alpha$
 - ▶ Если α имеет более одного вхождения символа B , то каждый символ B_k во всех β соотнесен (верхним индексом) с конкретным вхождением B

Входной грамматикой назовем четверку (N, Σ, P', S) , где $P' = \{A \rightarrow \alpha \mid A \rightarrow \alpha, A_1 = \beta_1, \dots, A_n = \beta_n \in P\}$

Выход обобщенной СУ-схемы

- Для каждой внутренней вершины дерева, соответствующей нетерминалу A , с каждым A_i связывается одна цепочка
 - ▶ Такую цепочку назовем значением (трансляцией) символа A_i
- Каждое значение определяется подстановкой значений символов трансляции данного элемента $A_i = \beta_i$, определенных в прямых потомках вершины
- **Трансляция**, определяемая данной схемой — множество $\{(\alpha, \beta)\}$
 - ▶ α имеет дерево разбора в данной входной грамматике
 - ▶ β — значение выделенного символа S_k

Пример обобщенной СУ-схемы: дифференцирование

$$\begin{array}{lcl} E & \rightarrow & E + T \quad , \quad E_1 = E_1 + T_1 \\ & & , \quad E_2 = E_2 + T_2 \\ & | & T \\ T & \rightarrow & T * F \quad , \quad E_1 = T_1, E_2 = T_2 \\ & & , \quad T_1 = T_1 * F_1 \\ & & , \quad T_2 = T_1 * F_2 + T_2 * F_1 \\ & | & F \\ F & \rightarrow & (E) \quad , \quad T_1 = F_1, T_2 = F_2 \\ & & , \quad F_1 = (E_1) \\ & & , \quad F_2 = (E_2) \\ & | & \sin(E) \quad , \quad F_1 = \sin(E_1) \\ & & , \quad F_2 = \cos(E_1) * E_2 \\ & | & \cos(E) \quad , \quad F_1 = \cos(E_1) \\ & & , \quad F_2 = -\sin(E_1) * E_2 \\ & | & x \quad , \quad F_1 = x, F_2 = 1 \\ & | & n \quad , \quad F_1 = n, F_2 = 0 \end{array}$$

Транслирующие грамматики

- КС-грамматика, терминальный алфавит которой разбит на два множества: входных и выходных символов
- **Транслирующая грамматика** — пятерка $(N, \Sigma_i, \Sigma_o, P, S)$
 - ▶ N — алфавит нетерминалов
 - ▶ Σ_i — алфавит входных терминалов
 - ▶ Σ_o — алфавит выходных терминалов
 - ▶ $S \in N$ — стартовый нетерминал
 - ▶ $P = \{A \rightarrow \alpha\}, \alpha \in (\Sigma_i \cup \Sigma_o \cup N)^*$ — множество правил вывода

Пример транслирующей грамматики

$$\begin{array}{lcl} E & \rightarrow & E + T \{+\} \\ & | & T \\ T & \rightarrow & T * F \{*\} \\ & | & F \\ F & \rightarrow & n \{n\} \\ & | & (E) \end{array}$$

Пример транслирующей грамматики

$$\begin{array}{lcl} E & \rightarrow & E + T \{+\} \\ & | & T \\ T & \rightarrow & T * F \{*\} \\ & | & F \\ F & \rightarrow & n \{n\} \\ & | & (E) \end{array}$$

$$\begin{aligned} E &\Rightarrow E + T \{+\} \Rightarrow T + T \{+\} \Rightarrow P + T \{+\} \Rightarrow n \{n\} + T \{+\} \Rightarrow \\ n \{n\} + T * P \{*\} \{+\} &\Rightarrow n \{n\} + P * P \{*\} \{+\} \Rightarrow \\ n \{n\} + n \{n\} * P \{*\} \{+\} &\Rightarrow n \{n\} + n \{n\} * n \{n\} \{*\} \{+\} \end{aligned}$$

Пример транслирующей грамматики

$$\begin{array}{lcl} E & \rightarrow & E + T \{+\} \\ & | & T \\ T & \rightarrow & T * F \{*\} \\ & | & F \\ F & \rightarrow & n \{n\} \\ & | & (E) \end{array}$$

$$\begin{aligned} E &\Rightarrow E + T \{+\} \Rightarrow T + T \{+\} \Rightarrow P + T \{+\} \Rightarrow n \{n\} + T \{+\} \Rightarrow \\ n \{n\} + T * P \{*\} \{+\} &\Rightarrow n \{n\} + P * P \{*\} \{+\} \Rightarrow \\ n \{n\} + n \{n\} * P \{*\} \{+\} &\Rightarrow n \{n\} + n \{n\} * n \{n\} \{*\} \{+\} \end{aligned}$$

- Если вычеркнуть все выходные символы, получим $n + n * n$
- Если вычеркнуть все входные символы, получим $nnn + *$ — постфиксная запись выражения

Постфиксная транслирующая грамматика

- Если выходные символы встречаются только в конце правил, транслирующая грамматика называется постфиксной
- Это требование формально не выдвигается: транслирующие грамматики могут быть не постфиксными
- На практике постфиксные транслирующие грамматики удобнее

Атрибутная транслирующая грамматика

- Входной алфавит — алфавит лексем
 - ▶ Лексема характеризуется **типом** и **значением**
- Транслирующая грамматика описывает перевод только **типа** лексем
 - ▶ Это существенно снижает выразительность формализма
- Для борьбы с этим недостатком предложены атрибутные грамматики
 - ▶ Модификация транслирующих грамматик, снабженная атрибутами
 - ▶ Выходные символы транслирующих грамматик — транслирующие символы
 - ★ Нетерминалы, которые раскрываются в ϵ , и в момент раскрытия выполняют связанные с ними действие

Атрибут — дополнительные данные, ассоциированные с грамматическими символами

- Если X — символ, а a — его атрибут, то значение a в узле дерева, помеченном X , записывается как $X.a$
- Узлы дерева могут реализовываться как записи или объекты, а атрибуты — как поля
- Атрибуты могут быть любого типа
- Если в каждом узле дерева атрибуты уже вычислены, оно называется **аннотированным**
- Процесс вычисления этих атрибутов называется **аннотированием** дерева разбора

Вычисление атрибутов не всегда возможно

$$\begin{array}{ll} A \rightarrow B & A.s = B.i \\ & B.i = A.s + 1 \end{array}$$

Синтезируемый атрибут, S-атрибутная грамматика

- Атрибут, значение которого зависит от значений атрибутов детей данного узла или от других атрибутов этого узла, называется **синтезируемым**
- Если в транслирующей грамматике используются только синтезируемые атрибуты, она называется **S-атрибутной грамматикой**
- Аннотирование дерева разбора S-атрибутной грамматики возможно путем выполнения семантических правил снизу вверх (от листьев к корню)

Пример S-атрибутной грамматики

$S \rightarrow E$ $S.val = E.val$

$E_0 \rightarrow E_1 + T \quad \{ADD.res = op_1 + op_2\}$ $ADD.op_1 = E_1.val$
 $ADD.op_2 = T.val$
 $E_0.val = ADD.res$

$E \rightarrow T$ $E.val = T.val$

$T_0 \rightarrow T_1 * F \quad \{MUL.res = op_1 * op_2\}$ $MUL.op_1 = T_1.val$
 $MUL.op_2 = F.val$
 $T_0.val = MUL.res$

$T \rightarrow F$ $T.val = F.val$

$F \rightarrow n$ $F.val = n.val$

$F \rightarrow (E)$ $F.val = E.val$

Наследуемый атрибут, L-атрибутная грамматика

Атрибут, значение которого зависит только от атрибутов братьев узла слева или атрибутов родителей, называется **наследуемым**

Грамматика называется **L-атрибутной**, если каждый наследуемый атрибут узла X_j в правиле $A \rightarrow X_1 \dots X_n$ зависит только от:

- Атрибутов узлов $X_1 \dots X_{j-1}$ (братья слева)
- Наследуемых атрибутов узла (предок)

Синтезируемые атрибуты тоже разрешены

Любая S-атрибутная грамматика является L-атрибутной

Пример L-атрибутной грамматики

$D \rightarrow TL$

$T \rightarrow int$

$T \rightarrow real$

$L.inh = T.type$

$T.type = integer$

$T.type = real$

$L_0 \rightarrow L_1, id \quad \{ENTRY.type = (k, v)\}$

$L_1.inh = L_0.inh$

$ENTRY.k = id.text$

$ENTRY.v = L_0.inh$

$L \rightarrow id \quad \{ENTRY.type = (k, v)\}$

$ENTRY.k = id.text$

$ENTRY.v = L.inh$