Лекции по Методам Трансляции 5 семестр

Ilya Yaroshevskiy

13 мая 2023 г.

Оглавление

Лекция 1

1.1 Введение

1. Лексический анализ
   Разбиение входа на **токены** (лексемы)

2. Синтаксический разбор (Parsing)
   По токенам и синтаксису языка получаем **дерево разбора**

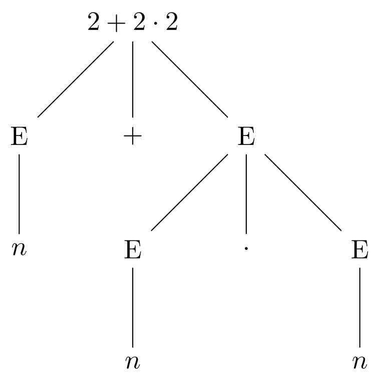
3. Компиляция
   По дереву разбора и семантике языка получаем **результат**

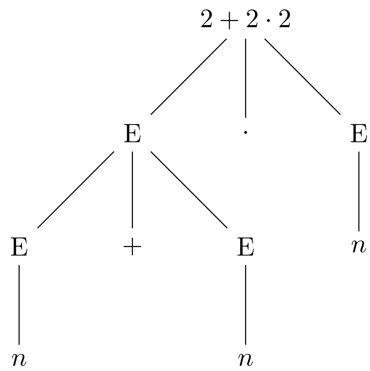
Пример.

- Токены: $\cdot, +, (,), n$(число)

- Грамматика
$$E \to n \mid (E) \mid E + E \mid E \cdot E$$

  Она неоднозначна:





Починим грамматику так, чтобы она была однозначна

$$\begin{array}{lcl}
E & \to & T \\
E & \to & T + E \text{ — Будет верно только для правовссоциативных} \\
E & \to & E + T \\
E & \to & E - T \\
T & \to & F \\
T & \to & T \cdot F \\
F & \to & n \\
F & \to & (E)
\end{array}$$

Часто удобно объединить шаги 2 и 3. В этом примере можно сразу вычислять значение выражения обходя построение дерева разбора.

Определение. Синтаксически управляемая трансляция — Доделать

1.2 Нисходящий анализ

Примечание. **КСГ** — способ задания КС языка

- Алфавит Σ — множество токенов
- Нетерминалы N
- Стартовый нетерминал $S \in N$
- Правила $P \subset N \times (N \cup \Sigma)^*$

  Обозначение. $\langle A, \alpha \rangle \in P$ — $A \to \alpha$

  Обозначение.

  - $\alpha \in (\sigma \cup N)^*$
  - $\alpha = \alpha_1 A \alpha_2$
  - $\beta = \alpha_1 \xi \alpha_2$
  - $A \to \xi \in P$

  $\alpha \Rightarrow \beta$ — из α выводится β за один шаг

- $L(\Gamma) = \{x \big| S \Rightarrow^* x\} \quad x \in \Sigma^*$
- Однозначная КСГ — для слова из языка единственное дерево разбора.

Определение. $\Gamma \in LL(1)$, если

- $S \Rightarrow^* xA\alpha \Rightarrow x\xi\alpha \Rightarrow^* xcy$
- $S \Rightarrow^* xA\beta \Rightarrow x\eta\beta \Rightarrow^* xcz$

$c \in \Sigma$ или $(c, y, z = \varepsilon)$
Тогда $\xi = \eta$

Определение. $\Gamma \in LL(k)$, если

- $S \Rightarrow^* xA\alpha \Rightarrow x\xi\alpha \Rightarrow^* xcy$
- $S \Rightarrow^* xA\beta \Rightarrow x\eta\beta \Rightarrow^* xcz$

$c \in \Sigma^k$ или $(c \in \Sigma^{<k}\ y, z = \varepsilon)$
Тогда $\xi = \eta$

Определение.

- $FIRST : (N \cup \Sigma)^* \to 2^{\Sigma \cup \{\varepsilon\}}$

$$FIRST(\alpha) \overset{def}{=\!=} \{c \big| \alpha \Rightarrow^* c\beta\} \cup \{\varepsilon \big| \alpha \Rightarrow^* \varepsilon\}$$

- $FOLLOW : N \to 2^{\Sigma \cup \{\$\}}$

$$FOLLOW(a) \overset{def}{=\!=} \{c \big| S \Rightarrow^* aAc\beta\} \cup \{\$ \big| S \Rightarrow^* aA\}$$

Теорема 1.2.1. $\Gamma \in LL(1) \Leftrightarrow \forall A \to \alpha,\ A \to \beta$

1. $FIRST(\alpha) \cap FIRST(\beta) = \varnothing$
2. $\varepsilon \in FIRST(\alpha) \Rightarrow FIRST(\beta) \cap FOLLOW(A) = \varnothing$

Лекция 2

2.1 $LL(1)$ грамматика

Лемма 1.

$$\alpha = c\beta \implies \text{FIRST}(\alpha) = \{c\}$$

$$\alpha = \varepsilon \implies \text{FIRST}(\alpha) = \{\varepsilon\}$$

$$\alpha = A\beta \implies \text{FIRST}(\alpha) = \text{FIRST}(\alpha) \setminus \varepsilon \cup (\text{FIRST}(\beta) \textit{ если } \varepsilon \in \text{FIRST}(A))$$

- *Массив* `FIRST[]: map<NonTerm, Set<Term1>>`: $N \to 2^{\Sigma \cup \{\varepsilon\}}$

```
while(change):
    for A → α ∈ Γ:
        FIRST[A] ∪= FIRST(α)
```

Доказательство. Исправить

$$\text{FIRST}[A] \subset \text{FIRST}(A) \implies c \in \text{FIRST}(A) \text{ и } c \notin \text{FIRST}[A]$$

Тогда $A \Rightarrow^k c\varepsilon$, $k \rightsquigarrow \min$. Предположим, что на k-той итерации ничего не поменялось, но мы прошли правило $A \Rightarrow^k \alpha \Rightarrow c\varepsilon$, **противоречие**. $\square$

Лемма 2.

- `FOLLOW[]`

```
FOLLOW[S] = \{$\}
while(change):
    for A → α ∈ Γ:
        for β : α = βBγ:
            FOLLOW[B] ∪= FIRST(γ)\ε ∪ (FOLLOW(A) если ε ∈ FIRST(γ))
```

Пример.

$$\begin{aligned}
&E \to E + T \\
&E \to T \\
&T \to T \cdot E \\
&T \to F \\
&F \to (E) \\
&F \to n
\end{aligned}$$

	FIRST	FOLLOW
E	$($ n	$\$ +)$
T	$($ n	$\$ + \cdot)$
F	$($ n	$\$ + \cdot)$

Проверим, что $LL(1)$:

$$\mathrm{FIRST}(E) \cap \mathrm{FIRST}(T) = \{(, n\} \neq \varnothing$$

$\implies$ **не** $LL(1)$

Доказательство Теоремы. Пусть не $LL(1)$, тогда $\xi \neq \eta$

1. ξ не порождает ε, η не порождает $\varepsilon \implies c \in \mathrm{FIRST}(\xi)$ и $c \in \mathrm{FIRST}(\eta)$
2. ξ порождает ε, η порождает $\varepsilon \implies \varepsilon \in \mathrm{FIRST}(\xi)$ и $\varepsilon \in \mathrm{FIRST}(\eta)$
3. ξ порождает ε, η не порождает $\varepsilon \implies \varepsilon \in \mathrm{FIRST}(\xi),\ c \in \mathrm{FOLLOW}(A),\ c \in \mathrm{FIRST}(\eta)$

□

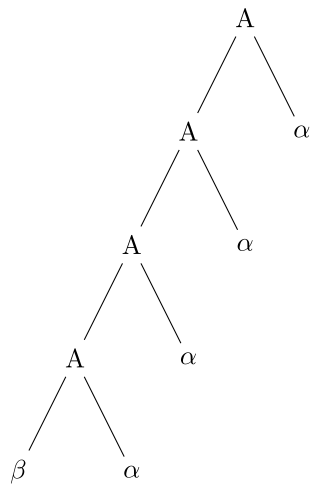
Определение. Γ — **леворекурсивная**, если

$$\exists A : A \Rightarrow^* A\alpha$$

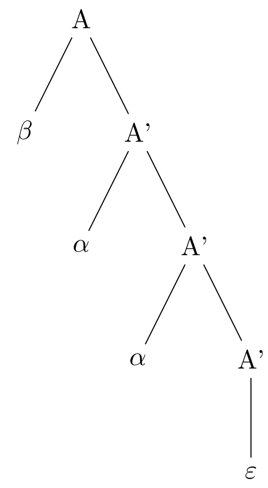
Утверждение. *Леворекурсивная* $\implies$ *не* $LL(1)$ *(почти всегда)*

Примечание. Непосредственная левая рекурсия: $A \to A\alpha$. A не порождающий, значит есть $A \to \beta$.

- $c \in \mathrm{FIRST}(\beta)$
- $c \in \mathrm{FIRST}(A\alpha)$



$$\Downarrow$$



$$
\begin{aligned}
&A \to \beta A' \\
&A' \to \alpha A' \\
&A' \to \varepsilon
\end{aligned}
$$

Определение. $A \to \alpha\beta$, $A \to \alpha\gamma$ — **правое ветвление**

Утверждение. *Правое ветвление* $\implies$ *не* $LL(1)$ *(почти всегда)*

$$
\begin{aligned}
&A \to \alpha A' \\
&A' \to \beta \\
&A' \to \gamma
\end{aligned}
$$

Пример. Перестроим грамматику для выражений:

$$
\begin{aligned}
&E \to TE' \\
&E \to +TE' \\
&E' \to \varepsilon \\
&T \to FT' \\
&T' \to \cdot FT' \\
&T' \to \varepsilon \\
&F \to (E) \\
&F \to n
\end{aligned}
$$

	FIRST	FOLLOW
E	n (	\$)
E'	ε +	\$)
T	n c	+ \$)
T'	ε $\cdot$	+ \$)
F	n c	+ $\cdot$ \$)

Это $LL(1)$ грамматика.

2.2 Конструирование парсеров

- Сверху (Top Down). Знаем корень, начинаем с него
- Снизу (Bottom Up). Знаем крону, начинаем с токенов

2.2.1 Рекурсивный спуск

Контекст:

- Текущий токен (`token`)
- `nextToken()` — следующий токен
- `FIRST1`$(\alpha) = \text{FIRST}(\alpha) \setminus \varepsilon \cup (\text{FOLLOW}(A)$ если $\varepsilon \in \text{FIRST}(\alpha))$

```
Tree A():
    r = Tree(A)
    switch(token):
        case FIRST1(α₁):
            // α₁
        case FIRST1(α₂):
            // α₂
        ⋮
        case FIRST1(αⱼ):
            for Xⱼ : αᵢ = X₁X₂...:
                // Xᵢ = c - терминал
                ensure Xⱼ = token
                nextToken()
                r.addChild(Xⱼ)
                // Xⱼ = A - нетерминал
                r.addChild(Xⱼ())
```

Лекция 3

3.1 Хвостовая рекурсия

Пример.

$$\begin{gathered} \to A\alpha \\ A \to \beta \\ \Downarrow \\ \to \beta A' \\ A' \to \alpha A' \\ A' \to \varepsilon \end{gathered}$$

```
def A():
    res = Node(A)
    switch(token):
        case FIRST1(β_1):
            # ⋮
        case FIRST1(β_2):
            # ⋮
        # ⋮
        cur = Node(A)
        res.add(cur)
        while token ∈ FIRST1(α_i):
            switch(token):
                cur.add(...)
                # ⋮
```

3.2 Управляющая таблица

$\Gamma - LL(1)$

Доделать

Пример.

$$
\begin{array}{ll}
(1) & E \to E' \\
(2) & E' \to +TE' \\
(3) & E' \to \varepsilon \\
(4) & T \to FT' \\
(5) & T' \to \cdot FT' \\
(6) & T' \to \varepsilon \\
(7) & F \to n \\
(8) & F \to (E)
\end{array}
$$

	FIRST	FOLLOW
E	$(, n$	$), \$$
E'	$+, \varepsilon$	$), \$$
T	$(, n$	$+,), \$$
T'	$\cdot, \varepsilon$	$,), \$$
F	$(, n$	$\cdot, +,), \$$

	n	$+$	$\cdot$	$($	$)$	$\$$
E	1			1		
E'		2			3	3
T	4		4			
T'		6	5		6	6
F	7			8		

```python
def E() -> int:
    res = T()
    while token == '+':
        nextToken()
        res += T()
    return res
```

```python
 def E() -> int:
     res = T()
     res = E_(res)
     return res
def E_(acc: int) -> int:
    if token == '+':
        nextToken()
        t = T()
        acc += t
        res = E_(acc)
    elif token == ')' or token == '$':
        res = acc
    return res
```