Введение

Определение *Грамматика* $G =< \Sigma >$

pos = init + delta * 60;

1. Лексический анализ
   <id, 1> <=> <id, 2> <+> <id, 2> <*> <const> <;>
2. Синтаксический анализ (разложение в дерево)
3. Семантический анализ
4. Промежуточное представление
   $t_1 = delta * 60$
   $t_2 = init + t_1$

Иерархия Хомского

малые латинские буквы - терминалы
большие - нетерминалы

Вид грамматики	Правила	Распознаватель	Класс языков
Общего вида (неограниченные)	$\alpha \to \beta$	МТ	RecEn
Контекстно-зависимые(КЗ)	$\alpha A \beta \to \alpha \gamma \beta$	Линейный ограниченный автомат (LBA)	КЗЯ
Контекстно-свободные(КС)	$A \to \beta$	Недетерминированный автомат с магазинной памятью (PDA)	КСЯ
Праволинейные	$A \to \gamma B$	ДКА	Регулярные

Пример:
$S \to ASB|\lambda$
$AB \to BA$
$A \to a$
$B \to b$
Эквивалентная грамматика:
$S \to aB|bA$
$A \to aS|bAA$
$B \to bS|aBB$
$A \to a$
$B \to b$
Значит исходная грамматика - праволинейная.

Регулярные $\subset$ КСЯ $\subset$ КЗЯ $\subset$ RecC $\subset$ RecEn

Определение *Язык обладает св-м* P*, если* $\exists$ *грамматика со св-м* P*, его порождающая*

КСГ и КСЯ

Определение *Упорядоченное дерево - дерево с заданным линейным порядком со св-ми:*

1. *если* x *- сын* y*, то* $x \geq y$
2. *если* x *и* y *- братья и* $x \leq y$*, то для всех сыновей* z *узла* x*:* $z \leq y$

Определение *Дерево вывода цепочки* w *в грамматике* G *- упорядоченное дерево со св-ми:*

1. *Узлы - нетерминалы, корень - акисиома, листья - терминалы или* λ*, причем у листьев* λ *нет братьев*
2. *Если у узла* x *сыновья* $y_1 \leq ... \leq y_n$*, то существует правило вывода* $X \to Y_1...Y_n$
3. *Если все листья дерева имеют метки* $a_1 \leq ... \leq a_n$*, то* $w = a_1...a_n$

Определение *Вывод цепочки* w *(*$S \to \alpha_1 \to ... \to \alpha_n = w$*) в* G *представлен деревом* T*, если существует набор стандартных поддеревьев* $T_1...T_n$ *такой, что упорядоченные листья* T_i *являются* α_i

Определение T' *- стандартное поддерево* T*, если:*

1. *Корни* T *и* T' *совпадают*
2. *если узел лежит в* T'*, то он либо лист в* T'*, либо все его сыновья лежат в* T'

Одной цепочке могут соответствовать несколько деревьев.

Определение *Грамматика однозначна, если любая цепочка имеет единственное дерево вывода. Язык однозначен, если существует порождающая его однозначная грамматика.*

Праволинейная грамматика

$A \to \alpha B$
$A \to \lambda$

Теорема *Праволинейная грамматика порождает регулярный язык.*

Доказательство:
$G = (\Sigma, \Gamma, P, S)$
$A = (\Sigma, \Gamma, \delta, S, F)$
$F = \{A \in \Gamma | (A \to \lambda) \in P\}$
$\delta(A, a) = B \Leftrightarrow (A \to aB) \in P$

Преобразования грамматик

1. Приведенные грамматики.

   Определение *Нетерминал* $A \in \Gamma$ *- производящий, если из него можно получить терминальную цепочку.*

   Определение *Нетерминал* $A \in \Gamma$ *- достижимый, если* $S \Rightarrow \alpha A \beta$

   Определение *Грамматика - приведенная, если все ее нетерминалы достижимые и проиводящие.*

   Алгоритм нахождения Γ_r - мн-ва производящих символов: $\Gamma \leftarrow S$

   $\Gamma_1 = \Gamma \cup \{A | (B \to \alpha A \beta) \in P, B \in \Gamma\}$

   Алгоритм нахождения Γ_p - мн-ва достижимых символов: $\Gamma = \{A | (A \to w) \in P\}$
   $\Gamma_1 = \Gamma \cup \{A | (A \to \gamma) \in P, \gamma \in (\Sigma \cup \Gamma)\}$

   Теорема *Для любой КСГ G существует эквивалентная ей приведенная грамматика.*

   Доказательство:
   $G = (\Sigma, \Gamma, P, S)$

   Находим Γ_p. *Если* $S \notin \Gamma_p$, *то* $G' = (\Sigma, \varnothing, \varnothing, \varnothing)$. *Иначе* $\overset{\triangle}{\Gamma} = (\Sigma, \Gamma_p, p', S)$

   $\overset{\triangle}{P} = \{(A \to \gamma) \in P | A, \gamma \in (\Sigma \cup \Gamma_p)\}$

   Находим $(\Gamma_p)_r$. *Все символы будут достижимые и производящие в* G'. *Порядок важен.*

2. λ-сведение грамматики.

   Определение $A \in \Gamma$ *- аннулирующий, если* $A \Rightarrow \lambda$

Алгоритм нахождения $Ann(G)$ - мн-ва аннулирующих нетерминалов.
$Ann(G) = \{A \in \Gamma | (A \to \lambda) \in P\}$ $Ann_1(G) = \{A \in \Gamma | (A \to \gamma) \in P, \gamma \in (Ann(G))*\}$

Определение λ *- свободная грамматика - грамматика, которая либо не содержит аннулирующих правил* $(A \to \lambda)$, *либо содержит единственное такое правило из* S *и* S *не встречается в правых частях правил вывода.*

Теорема *Любая грамматика эквивалентна некоторой* λ*-свободной грамматике*

Доказательство:
$G = (\Sigma, \Gamma, P, S)$
Если $\lambda \in L(G)$, *то* $\Gamma' = \Gamma \cup S'$, $P' = P \cup \{(S' \to \lambda), (S' \to S)\}$. *Иначе не изменяются.*
Построим $Ann(G)$.
$\beta \prec \gamma$, *если* β *- подпослед-ть* γ *и все символы* $\gamma \setminus \beta$ *- аннулирующие.*
$P' = \{(A \to \beta) | (A \to \gamma) \in P, \beta \prec \gamma, \beta \neq \lambda\}$
Проверим, что $L(G) = L(G')$.

1. $w \in L(G')$, $S \to \alpha_1 \to ... \to \alpha_n = w$

   Если $A \to_{G'} \beta$, *то* $A \to_G \gamma$, *где* $\beta \prec \gamma$

2. $w \in L(G)$

   ...