



лекция 24

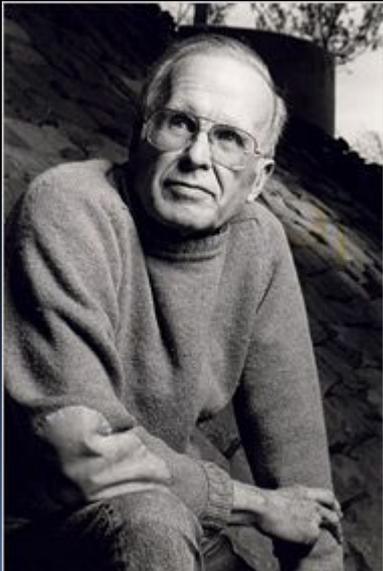
ЭЛЕМЕНТЫ ТЕОРИИ ЯЗЫКОВ

План лекции

- Описание синтаксиса языков
 - БНФ, РБНФ, синтаксические диаграммы
- Формальные грамматики
- Классификация грамматик по Хомскому
- Распознавание языков
 - Синтаксический анализатор, нис- и восходящий разбор, полный перебор правил подстановки
 - Определение языков с помощью автоматов

Форма Бекуса-Наура описания синтаксиса формальных языков

- Джон Бекус (John Backus, 1924-2007)
 - Руководил созданием первого компилятора для языка Фортран
- Питер Наур (Peter Naur, 1925)
 - Один из создателей языка Алгол
 - "Backus Normal Form"



Форма Бекуса-Наура описания синтаксиса формальных языков

- Описание синтаксиса языков программирования
- Терминальные символы
- Нетерминальные символы
- Правила вида
 - $\langle \text{нетерм.символ} \rangle ::= \langle \text{посл.симв.1} \rangle$
| $\langle \text{посл.симв.2} \rangle$
| ...
| $\langle \text{посл.симв.n} \rangle$

Пример БНФ № 1

- $\langle \text{цифра} \rangle ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'$
- $\langle \text{знак} \rangle ::= '+' | '-' |$
- $\langle \text{число без знака} \rangle ::= \langle \text{цифра} \rangle | \langle \text{цифра} \rangle \langle \text{число без знака} \rangle$
- $\langle \text{число} \rangle ::= \langle \text{знак} \rangle \langle \text{число без знака} \rangle$
- Множество строк, которые описывает $\langle \text{число} \rangle$:
 - 0, 1, ..., 9, +0, +1, ..., +9, -0, -1, ..., -9, 00, 01, ..., 09, +00, +01, ..., +09, -00, -01, ..., -09, ...

Пример БНФ № 2

- Какое множество строк описывает $\langle \text{ппс} \rangle$?
- $\langle \text{ппс} \rangle ::= | '(' \langle \text{ппс} \rangle ')' | \langle \text{ппс} \rangle \langle \text{ппс} \rangle$

Пример БНФ № 3

- Опишите БНФ при помощи БНФ

Расширенная БНФ

- [\langle ПОСЛ.СИМВ. \rangle]
 - Необязательная последовательность СИМВОЛОВ
- { \langle ПОСЛ.СИМВ. \rangle }
 - Повторение последовательности СИМВОЛОВ

Грамматика

- Формальный язык – это произвольное множество цепочек, составленных из символов некоторого конечного алфавита
 - Произвольное -- бесконечное, конечное или пустое
- Грамматика – это конечное описание формального языка

Определение грамматики

- Грамматика – это набор из четырех элементов
 - Множество терминальных символов
 - Алфавит языка
 - Множество нетерминальных символов
 - Вспомогательные символы, не входящие в описываемый язык
 - Множество правил вида $ЛЧ \rightarrow ПЧ$, где
 - ЛЧ – послед. терминалов и нетерминалов, содержащая ≥ 1 нетерминал
 - ПЧ – любая последовательность нетерминалов
 - Стартовый нетерминал S

Применение правил грамматики

- Цепочка ζ_2 получается из цепочки ζ_1 применением правила $ЛЧ \rightarrow ПЧ$, если ζ_1 имеет вид $x ЛЧ y$, а ζ_2 имеет вид $x ПЧ y$
- Пример
 - Цепочка $aaABvv$ получается из $aABv$ применением правила $AB \rightarrow aABv$

Вывод в грамматике

- Вывод цепочки ζ – это последовательность цепочек, состоящих из терминалов и нетерминалов, вида S, \dots, ζ , где каждая последующая цепочка получена из предыдущей путем применением одного (любого) правила грамматики
- Язык, описываемый грамматикой, – это множество цепочек терминальных символов, для которых есть вывод

Примеры грамматик

- Язык $\{a+a, a+a+a, a+a+a+a, \dots\}$
- $T = \{a, +\}$, $N = \{S, A\}$, стартовый символ S , правила
 1. $S \rightarrow aA$
 2. $A \rightarrow +aA$
 3. $A \rightarrow +a$
- Пример вывода $a+a+a$
 - $S \xrightarrow{p1} aA \xrightarrow{p2} a+aA \xrightarrow{p3} a+a+a$

Примеры грамматик

- Язык $\{a, aaaa, aaaaaaaaa, \dots\}$ – строки из n^2 символов a
- $T = \{a\}$, $N = \{S, S', A, B, C, L, R\}$, стартовый символ S , правила
 1. $S \rightarrow LS'R$
 2. $S' \rightarrow AS'B$
 3. $S' \rightarrow AB$
 4. $AB \rightarrow BAC$
 5. $AC \rightarrow CA$
 6. $CB \rightarrow BC$
 7. $LB \rightarrow L$
 8. $AR \rightarrow R$
 9. $LC \rightarrow aL$
 10. $LR \rightarrow$
- Пример вывода $aaaa$
- Порождаем LA^nB^nR
 - $S \xrightarrow{LS'R} LAS'BR \xrightarrow{LAABBR}$
- Несем B налево и порожаем C при переходе B через A – число C равно n^2
 - $LABACBR \xrightarrow{LBACACBR} LBACABCR \xrightarrow{LBACBACCR} LBABCACCR \xrightarrow{LBACCACCR}$
- Удаляем A и B
 - $LBACCACCR \xrightarrow{LACCACCR} LCACACCR \xrightarrow{LCCAACCR} LCCACACR \xrightarrow{LCCCACAR} LCCCACR \xrightarrow{LCCCCAR} LCCCCR$
- Заменяем C на a , удаляем L и R
 - $aLCCCCR \xrightarrow{aaLCCR} aaaLCR \xrightarrow{aaaaLR} aaaa$

Классификация грамматик по Хомскому

- Ноам Хомски (Ноум Чомски, Noam Chomsky), 1928
- Классификация (иерархия) грамматик по сложности распознавания описываемых ими языков



Классификация грамматик по Хомскому – тип 0

- Тип 0 – произвольные грамматики
 - Любое рекурсивно перечислимое множество можно описать как язык с грамматикой типа 0
 - Нетривиальный результат
 - Любой язык с грамматикой типа 0 является рекурсивно перечислимым множеством
 - Почему?
 - Есть языки с грамматикой типа 0, для которых проверка принадлежности алгоритмически неразрешима

Классификация грамматик по Хомскому – тип 1

- Тип 1 – контекстно-зависимые грамматики
 - $\alpha A \beta \rightarrow \alpha \gamma \beta$, где α, β произвольные цепочки, γ непустая цепочка, A нетерминал
- Правила можно привести к виду $\alpha \rightarrow \beta$, где α, β непустые цепочки и $1 \leq |\alpha| \leq |\beta|$
 - Неукорачивающие грамматики
- Принадлежность любой цепочки языку м.б. проверена алгоритмом
 - Аналог рекурсивных множеств

Классификация грамматик по Хомскому – тип 2

- Тип 2 – контекстно-свободные грамматики
 - $A \rightarrow \beta$, где β цепочка терминалов и нетерминалов, A нетерминал
 - Описание языков программирования
 - Эквивалентны БНФ
 - Автоматическая генерация алгоритмов распознавания
 - Рекурсивный спуск
 - Быстрые LL и LR парсеры для языков со специальными КС грамматиками

LL анализатор языка с КС грамматикой

- Лента
 - Входной буфер, он же анализируемая цепочка
- Стек
 - Промежуточные данные синтаксического анализа
- Таблица синтаксического анализа
 - Либо правило грамматики для символа на вершине стека и текущего символа на ленте
 - Либо пометка об отсутствии правила для такой пары символов

LL анализатор языка с КС грамматикой

- Грамматика $T=\{+,(,),1\}$, $N=\{S,F\}$, правила
 1. $S \rightarrow F$
 2. $S \rightarrow (S+F)$
 3. $F \rightarrow 1$
- Таблица ($\$$ -- вспомогательный терминал "конец стека")

	()	1	+	\$
S	п2	-	п1	-	-
F	-	-	п3	-	-

LL анализатор языка с КС грамматикой -- пример

Стек	Лента
S\$	(1+1)\$
(S+F)\$	(1+1)\$
S+F)\$	1+1)\$
F+F)\$	1+1)\$
1+F)\$	1+1)\$
+F)\$	+1)\$
F)\$	1)\$
1)\$	1)\$
)\$)\$
\$	\$

	()	1	+	\$
S	п2	-	п1	-	-
F	-	-	п3	-	-

LL анализатор языка с КС грамматикой

- Пока не конец
 - Вершина стека нетерминал
 - В таблице находим правило грамматики на пересечении столбца и строки, соответствующих нетерминалу на вершине стека и текущему символу на ленте, и кладем в стек цепочку из правой части правила
 - Если в указанной ячейке таблицы правило отсутствует, то сообщаем об ошибке
 - Вершина стека терминал
 - Сравниваем его с текущим символом на ленте
 - Если они равны, то удаляем символ с ленты и из стека
 - Иначе ошибка
 - Вершина \$
 - Текущий символ на ленте \$, то конец
 - Иначе ошибка

LL анализатор языка с КС грамматикой – построение

таблицы:

- $A \rightarrow aX$
- $A \rightarrow zAat$

Классификация грамматик по Хомскому – тип 3

- Тип 3 – регулярные грамматики
 - $A \rightarrow \gamma B$ или $A \rightarrow \gamma$, где γ цепочка терминалов, A и B нетерминалы
 - Правила можно привести к виду $A \rightarrow B\gamma$
 - Для любого языка с регулярной грамматикой можно построить конечный автомат, распознающий этот язык
 - Любой конечный автомат задает язык с регулярной грамматикой

Заключение

- Описание синтаксиса языков
 - БНФ, РБНФ, синтаксические диаграммы
- Формальные грамматики
- Классификация грамматик по Хомскому
- Распознавание языков
 - Нис- и восходящий разбор, полный перебор правил подстановки
 - Определение языков с помощью автоматов