Теория автоматов и формальных языков

Введение

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

6 сентября 2016г.

Контекст: языки

- Естественные
  - Русский, английский...

Контекст: языки

- Естественные
  - Русский, английский...
- Искусственные
  - Эсперанто, ложбан...
  - Клингонский, эльфийский...

Контекст: языки

- Естественные
  - Русский, английский...
- Искусственные
  - Эсперанто, ложбан...
  - Клингонский, эльфийский...
  - C++, Java, C#, Haskell, OCaml, Perl, Coq, Agda...

Контекст: языковые процессоры

- Текстовые редакторы
- Компиляторы, интерпретаторы, трансляторы
- Среды разработки

- Все нуждаются в некотором формализованном представлении языка

Язык программирования

- Синтаксис — правила построения программ из символов
- Семантика — правила истолкования программ, определяющие их смысл

Пример: язык арифметических выражений

- Алфавит символов: цифры, скобки, знаки арифметических операций $(+, -, *, /)$
- Синтаксис
  - **Терм**: последовательность цифр или любое **выражение** в скобках
  - **Слагаемое**: последовательность **термов**, соединненых знаками умножения и деления
  - **Выражение**: последовательность **слагаемых**, соединенных знаками сложения и вычитания (перед первым **слагаемым** может стоять минус)
- Семантика
  - Значение выражения

Метаязык

- Язык, на котором дано описание языка
  - Естественный язык

Метаязык

- Язык, на котором дано описание языка
  - Естественный язык
  - Язык металингвистичесих формул Бэкуса (БНФ)

Метаязык

- Язык, на котором дано описание языка
  - Естественный язык
  - Язык металингвистичесих формул Бэкуса (БНФ)
  - Синтаксические диаграммы

Метаязык

- Язык, на котором дано описание языка
  - Естественный язык
  - Язык металингвистичесих формул Бэкуса (БНФ)
  - Синтаксические диаграммы
  - Грамматики...

Алфавит

- **Алфавит** — конечное множество символов
  - $\{a, b, c, \ldots, z\}$
  - $\{\alpha, \beta, \gamma, \ldots, \omega\}$
  - $\{0, 1\}$
  - { let, in, where, ... }

Цепочка

- **Цепочка (предложение, слово)** — любая конечная последовательность символов алфавита
  - cat
  - $\kappa\alpha\tau$
  - 011000110110000101110100
  - `main = putStrLn . show . inc 2 where inc = \x -> x + 1`
- **Пустая цепочка** ε — цепочка, не содержащая ни одного символа
  - ε не является символом алфавита

Конкатенация строк

- **Конкатенация строк** α **и** β $(\alpha \cdot \beta = \alpha\beta)$ — результат приписывания строки β в конец строки α
  - $\forall \alpha\beta\gamma.(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$
  - $\forall \alpha.\alpha \cdot \varepsilon = \varepsilon \cdot \alpha = \alpha$

БНФ — Бэкуса-Наура форма

- **Символ** — элементарное понятие языка
  - $+$ означает сложение в языке арифметических выражений
- **Метапеременная** — сложное понятие языка
  - Переменной <выражение> можно обозначить выражение
- **Формула**
  - <определяемый символ>::=<посл.1 > $|\ldots|$ <посл.n >
  - В правой части формулы — альтернатива конкатенаций строк, составленных из символов и метапеременных
- Пример: число
  - <число>::=<цифра> | <цифра><число>

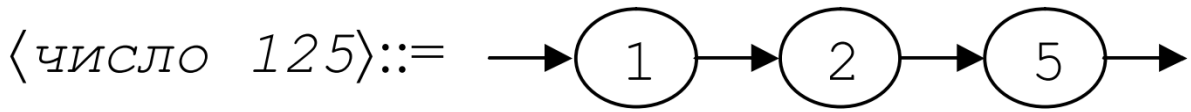
Расширенная форма Бэкуса Наура (EBNF)

- Более емкие операции
- **Итерация**
  - <x> ::= {<y>} эквивалентно: <x> ::= ε | <y><x>
- **Условное вхождение**
  - <x> ::= [<y>] эквивалентно: <x> ::= ε | <y>
- Скобки для группировки
  - (<x> | <y>) <z> эквивалентно: <x><z> | <y><z>

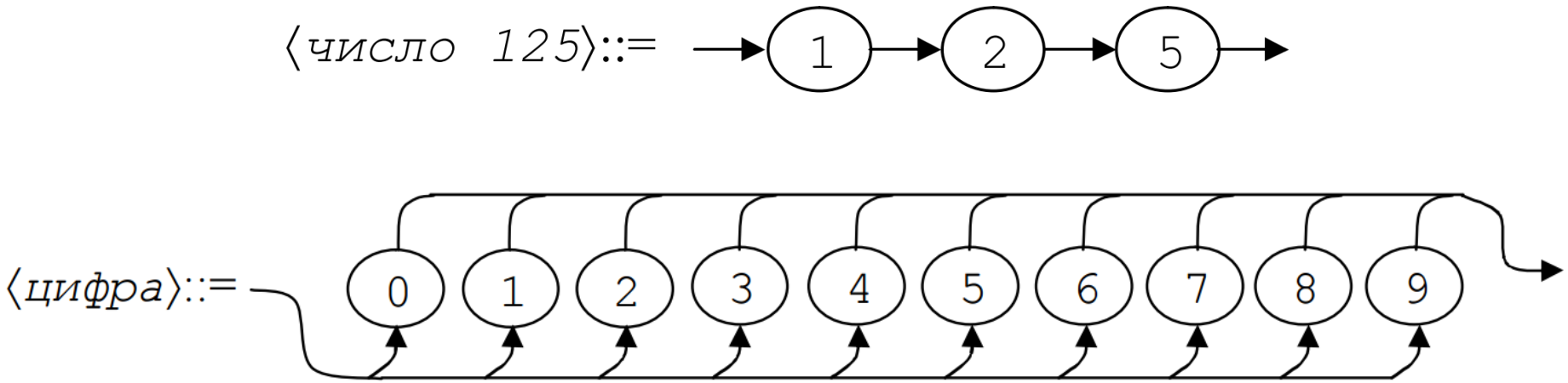
Пример: арифметические выражения

$$
\begin{aligned}
< \mathit{expr} > &::= [-] < \mathit{factor} > \{< +- >< \mathit{factor} >\} \\
< +- > &::= + \mid - \\
< \mathit{factor} > &::= < \mathit{term} > \{< */ >< \mathit{term} >\} \\
< */ > &::= * \mid / \\
< \mathit{term} > &::= < \mathit{number} > \mid (< \mathit{expr} >)
\end{aligned}
$$

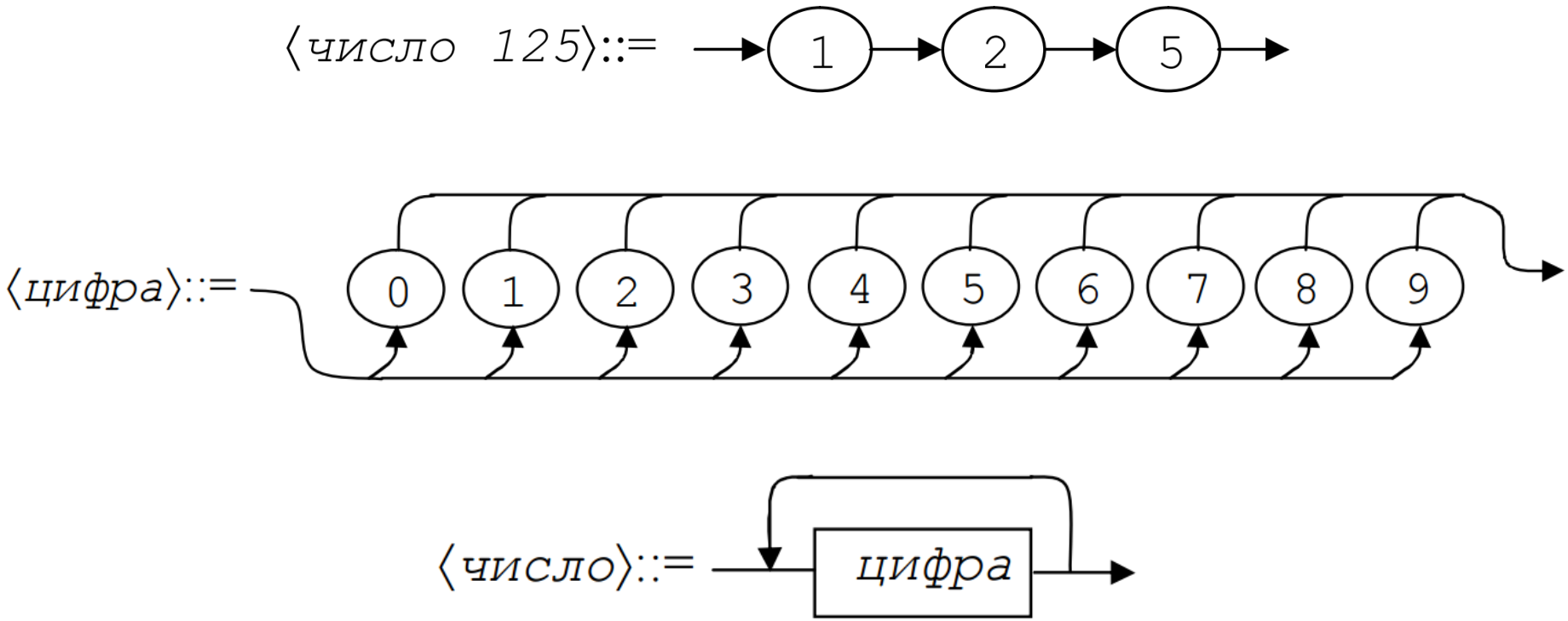
Синтаксические диаграммы Вирта



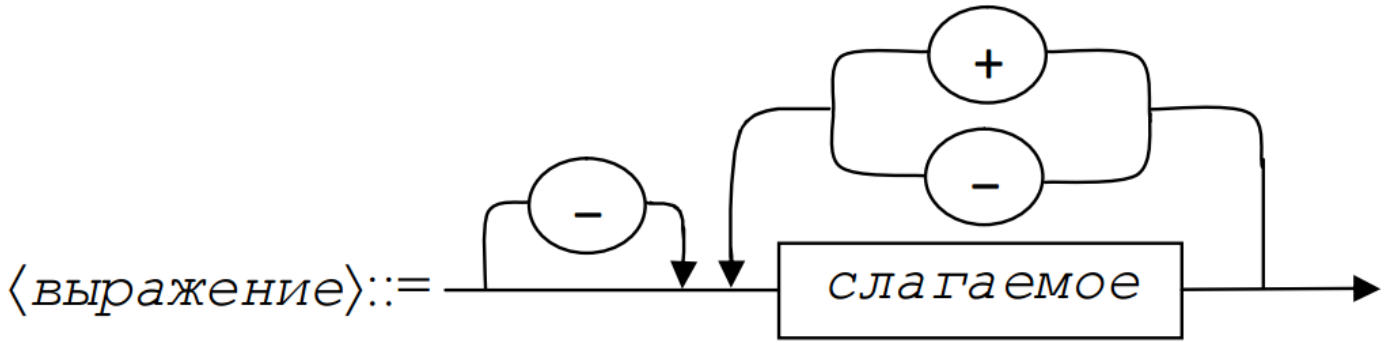
Синтаксические диаграммы Вирта



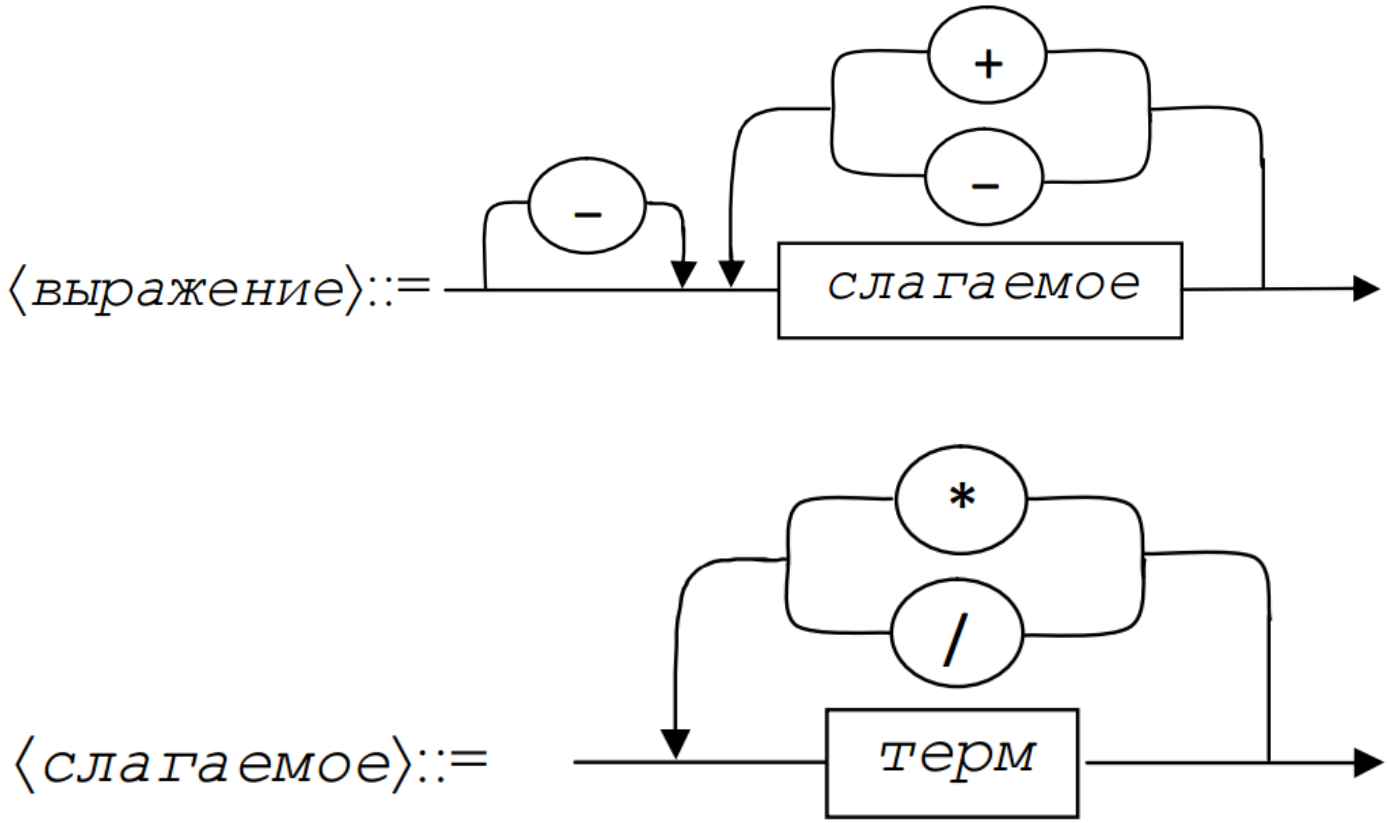
Синтаксические диаграммы Вирта



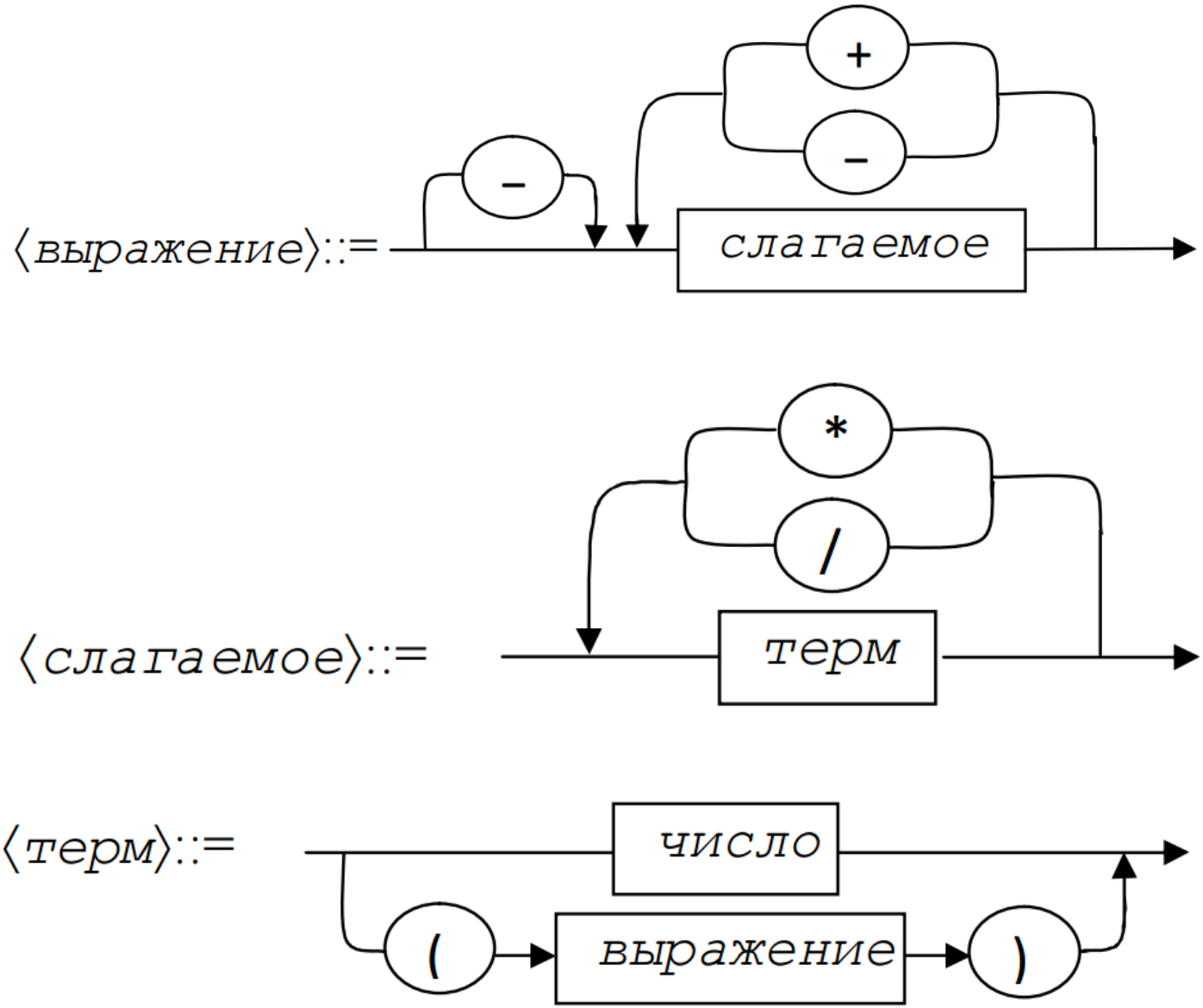
Синтаксические диаграммы Вирта



Синтаксические диаграммы Вирта



Синтаксические диаграммы Вирта



Операции над строками

- **Обращение (реверс) цепочки** a^R — цепочка, символы которой записаны в обратном порядке
  - Если $x = abc$, $x^R = cba$
  - $\varepsilon^R = \varepsilon$
- **$n-$я степень цепочки** a^n — конкатенация n повторений цепочки
  - $a^0 = \varepsilon$
  - $a^n = a \cdot a^{n-1} = a^{n-1} \cdot a$
- **Длина цепочки** $|a|$ — количество составляющих ее символов
  - $|babb| = 4, |babb|_a = 1, |babb|_b = 3, |babb|_c = 0$
  - $|\varepsilon| = 0$

Формальный язык

- V — алфавит
  - $V = \{0, 1\}$
- V^* — множество, содержащее все цепочки в алфавите V, включая пустую цепочку
  - $V^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, ...\}$
- $V^+ = V^* \setminus \{\varepsilon\}$
  - $V^+ = \{0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$
- V — подмножество множества всех цепочек в этом алфавите.
  - Для любого языка L справедливо $L \in V^*$

Описание языка: формальная грамматика

- **Порождающая грамматика** G — это четверка $\langle V_T, V_N, P, S \rangle$
  - V_T — алфавит терминальных символов (терминалов)
  - V_N — алфавит нетерминальных символов (нетерминалов)
    - $V_T \cap V_N = \emptyset$
    - $V ::= V_T \cup V_N$
  - P — конечное множество правил вида $\alpha \to \beta$
    - $\alpha \in V^* V_N V^*$
    - $\beta \in V^*$
  - S — начальный нетерминал грамматики, $S \in N$

Пример: язык чисел в двоичной системе счисления

$V_T = \{0, 1\}; V_N = \{S, N, A\}$

$$
\begin{array}{lcl}
S & \to & 0 \\
S & \to & N \\
S & \to & -N \\
N & \to & 1A \\
A & \to & 0A \\
A & \to & 1A \\
A & \to & \varepsilon
\end{array}
$$

Пример: язык чисел в двоичной системе счисления

$V_T = \{0, 1\}; V_N = \{S, N, A\}$

$$
\begin{array}{lcl}
S & \to & 0 \\
S & \to & N \\
S & \to & -N \\
N & \to & 1A \\
A & \to & 0A \\
A & \to & 1A \\
A & \to & \varepsilon
\end{array}
\qquad
\begin{array}{lcl}
S & \to & 0 \,|\, N \,|\, -N \\
N & \to & 1A \\
A & \to & 0A \,|\, 1A \,|\, \varepsilon
\end{array}
$$

Пример: язык чисел в двоичной системе счисления

$V_T = \{0, 1\}; V_N = \{S, N, A\}$

$$
\begin{array}{lcl}
S & \to & 0 \\
S & \to & N \\
S & \to & -N \\
N & \to & 1A \\
A & \to & 0A \\
A & \to & 1A \\
A & \to & \varepsilon
\end{array}
$$

$$
\begin{array}{lcl}
S & \to & 0 \mid N \mid -N \\
N & \to & 1A \\
A & \to & 0A \mid 1A \mid \varepsilon
\end{array}
$$

$$
\begin{array}{lcl}
S & \to & 0 \mid [-]N \\
N & \to & 1A \\
A & \to & (0 \mid 1)A \mid \varepsilon
\end{array}
$$

Отношение непосредственной выводимости

- $\alpha \to \beta \in P$
- $\gamma, \delta \in V^*$
- $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$: $\gamma\beta\delta$ **непосредственно выводится** из $\gamma\alpha\delta$ при помощи правила $\alpha \to \beta$

Отношение выводимости

- $a_0, a_1, a_2, \ldots, a_n \in V^*$
- $a_0 \Rightarrow a_1 \Rightarrow a_2 \Rightarrow \cdots \Rightarrow a_n$
- $a_0 \overset{*}{\Rightarrow} a_n$: a_n **выводится** из a_0
- $S \Rightarrow -N \Rightarrow -1A \Rightarrow -11A \overset{*}{\Rightarrow} -1101A \Rightarrow -1101$

Отношение выводимости

- $a_0, a_1, a_2, \ldots, a_n \in V^*$
- $a_0 \Rightarrow a_1 \Rightarrow a_2 \Rightarrow \cdots \Rightarrow a_n$
- $a_0 \overset{*}{\Rightarrow} a_n$: a_n **выводится** из a_0
- $S \Rightarrow -N \Rightarrow -1A \Rightarrow -11A \overset{*}{\Rightarrow} -1101A \Rightarrow -1101$
- $\forall a \in V^*.a \overset{*}{\Rightarrow} a$
- $a_0 \overset{+}{\Rightarrow} a_n$: вывод использует хотя бы одно правило грамматики
- $a_0 \overset{k}{\Rightarrow} a_n$: вывод происходит за k шагов

Язык, порождаемый грамматикой $G = \langle V_T, V_N, P, S \rangle$

- $L(G) = \{\omega \in V_T^* \mid S \stackrel{*}{\Rightarrow} \omega\}$

Эквивалентность грамматик

- Грамматики G_1 и G_2 **эквивалентны**, если $L(G_1) = L(G_2)$

Эквивалентность грамматик

- Грамматики G_1 и G_2 **эквивалентны**, если $L(G_1) = L(G_2)$

$$
\begin{aligned}
V_T &= \{0, 1\} \\
V_N &= \{S, N, A\}
\end{aligned}
$$

$$
\begin{aligned}
S &\to 0 \,|\, N \,|\, -N \\
N &\to 1A \\
A &\to 0A \,|\, 1A \,|\, \varepsilon
\end{aligned}
$$

Эквивалентность грамматик

- Грамматики G_1 и G_2 **эквивалентны**, если $L(G_1) = L(G_2)$

$$
\begin{aligned}
V_T &= \{0, 1\} \\
V_N &= \{S, N, A\} \\
\\
S &\to 0 \,|\, N \,|\, -N \\
N &\to 1A \\
A &\to 0A \,|\, 1A \,|\, \varepsilon
\end{aligned}
$$

$$
\begin{aligned}
V_T &= \{0, 1\} \\
V_N &= \{S, A\} \\
\\
S &\to 0 \,|\, 1A \,|\, -1A \\
A &\to 0A \,|\, 1A \,|\, \varepsilon
\end{aligned}
$$

Контекстно-свободная грамматика

- **Контекстно-свободная грамматика** — грамматика, все правила которой имеют вид $A \to \alpha, A \in V_N, \alpha \in V^*$

Дерево вывода

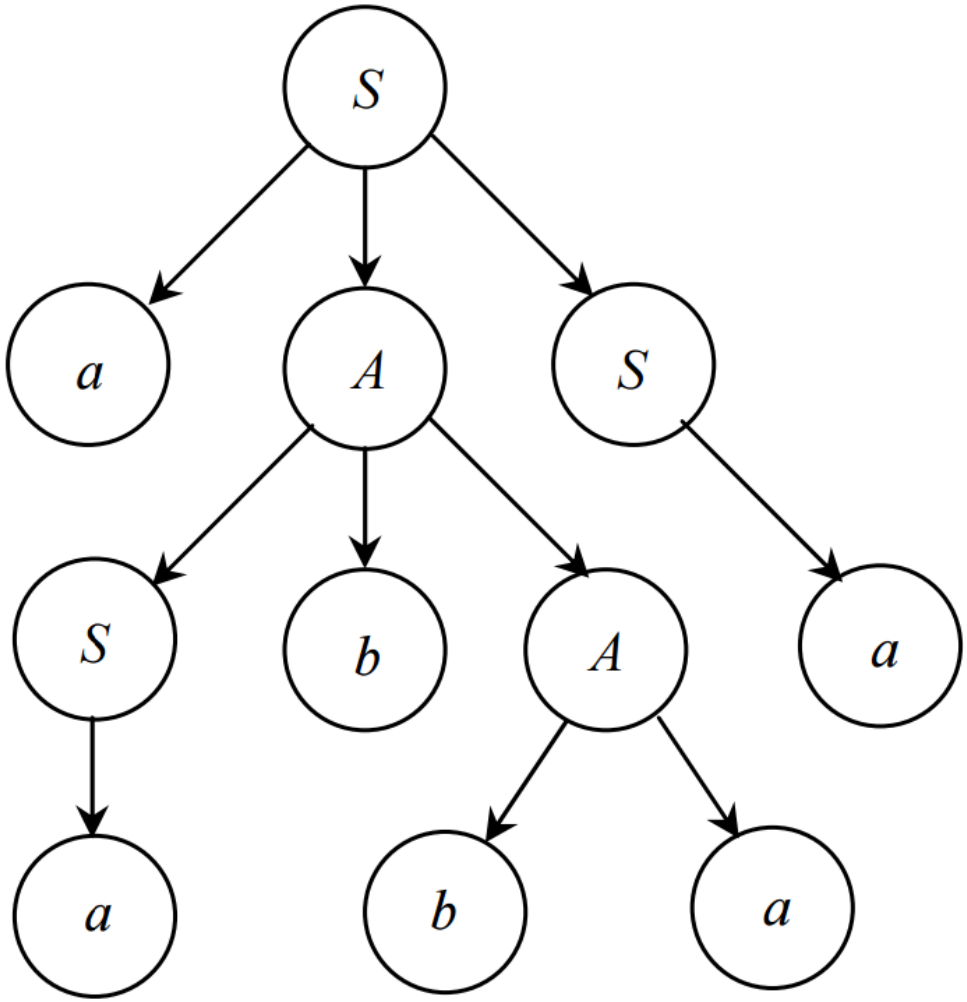
Дерево является **деревом вывода** для $G = \langle V_N, V_T, P, S \rangle$, если:

- Каждый узел помечен символом из алфавита V
- Метка корня — S
- Листья помечены терминалами, остальные узлы — нетерминалами
- Если узлы $n_0, \ldots, n_k$ — прямые потомки узла n, перечисленные слева направо, с метками $A_0, \ldots, A_k$; метка n — A, то $A \to A_0 \ldots A_k \in P$

Пример дерева вывода

$$G = \langle \{S, A\}, \{a, b\}, \{S \to aAS|a, A \to SbA|ba|SS\}, S \rangle$$

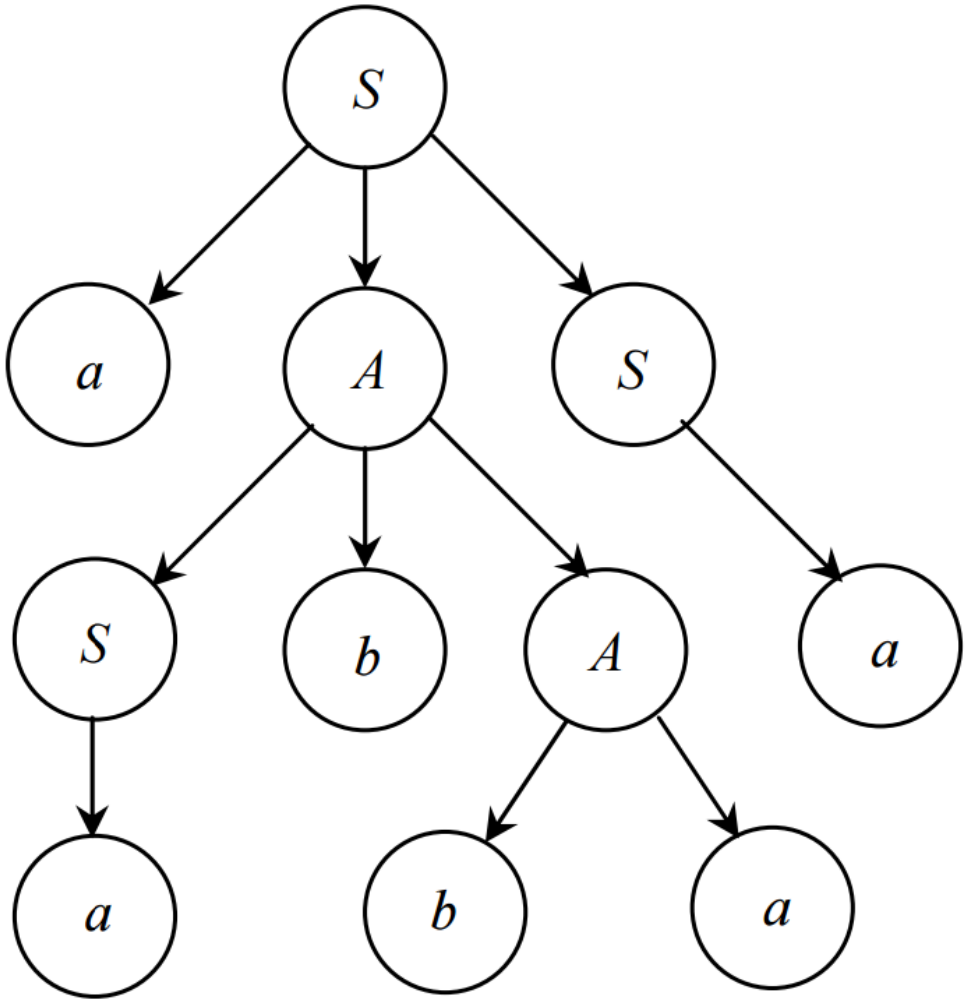
$$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$$



Пример дерева вывода

$$G = \langle \{S, A\}, \{a, b\}, \{S \to aAS|a, A \to SbA|ba|SS\}, S \rangle$$

$$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$$



Вывод и дерево вывода

Теорема

Пусть $G = \langle V_N, V_T, P, S \rangle$ *— КС-грамматика*
Вывод $S \overset{*}{\Rightarrow} \alpha$, *где* $\alpha \in V^*, \alpha \neq \varepsilon$ *существует* $\Leftrightarrow$ *существует дерево вывода в грамматике* G *с результатом* α