

Теория автоматов и формальных языков

Введение

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

5 сентября 2019

О чем этот курс?

Теория автоматов и формальных языков изучает:

- Математические модели для описания языков
- Абстрактные машины для работы с языками

Также рассматриваются:

- Подходы к описанию синтаксиса языков
- Подходы к описанию “смысла” программ и предложений
- Принципиальные ограничения механизмов для работы с языками

Какие бывают языки?

Какие бывают языки?

- Естественные
 - ▶ Русский, английский...

Какие бывают языки?

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

Какие бывают языки?

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

Где можно встретить языки?

В повседневной жизни:

- при разговоре, в переписке
- на заборах, на стенах гробниц
- в собственной голове при формулировке мыслей...

Где можно встретить языки?

В повседневной жизни:

- при разговоре, в переписке
- на заборах, на стенах гробниц
- в собственной голове при формулировке мыслей...

При работе с различными языковыми процессорами:

- текстовыми редакторами
- компиляторами, интерпретаторами, трансляторами
- средами разработки...

Где можно встретить языки?

В повседневной жизни:

- при разговоре, в переписке
- на заборах, на стенах гробниц
- в собственной голове при формулировке мыслей...

При работе с различными языковыми процессорами:

- текстовыми редакторами
- компиляторами, интерпретаторами, трансляторами
- средами разработки...

Все нуждаются в **формализованном представлении** языка

Два аспекта спецификации языка программирования

- Синтаксис — правила построения программ из символов
 - ▶ Форма
- Семантика — правила истолкования программ
 - ▶ Смысл

You know nothing, Jon Snow

- Синтаксис

- ▶ ...
- ▶ Порядок слов в предложении: подлежащее, дополнение сказуемое, все остальное
- ▶ Обращение обособляется запятыми
- ▶ ...

- Семантика

- ▶ Говорящий обращается к Джону Сноу, утверждая, что Джон ничего не знает

Пример: язык арифметических выражений

$$1 * (2 + 3) / 4 - 5$$

- Синтаксис

- ▶ **Терм**: последовательность цифр или любое **выражение** в скобках
- ▶ **Слагаемое**: последовательность **термов**, соединенных знаками умножения и деления
- ▶ **Выражение**: последовательность **слагаемых**, соединенных знаками сложения и вычитания (перед первым **слагаемым** может стоять минус)

- Семантика

- ▶ Значение арифметического выражения

Пример: язык арифметических выражений

$$1 * (2 + 3) / 4 - 5$$

- Синтаксис

- ▶ **Терм**: последовательность цифр или любое **выражение** в скобках
- ▶ **Слагаемое**: последовательность **термов**, соединенных знаками умножения и деления
- ▶ **Выражение**: последовательность **слагаемых**, соединенных знаками сложения и вычитания (перед первым **слагаемым** может стоять минус)

- Семантика

- ▶ Значение арифметического выражения
 - ★ -3.75
 - ★ -4

Пример: синтаксис if-выражений

```
if temperature > 23:  
    print('Wear shorts.')  
else:  
    print('Wear long pants.')
```

```
if ( temperature > 23 ) {  
    cout<<"Wear shorts.\n";  
}  
else  
    cout<<"Wear long pants.\n";  
}
```

```
if temperature > 23  
then print "Wear shorts."  
else print "Wear long pants."
```

```
(if (> temperature 23)  
  (print "Wear shorts.")  
  (print "Wear long pants."))
```

Что такое язык?

Что такое язык?

Язык — множество строк

Что такое множество?

Что такое множество?

Множество — набор уникальных элементов

Что такое множество?

Множество — набор уникальных элементов

- $x \in X$: x — элемент множества X (x принадлежит X)
- $x \notin X$: x не является элементом множества X (x не принадлежит X)
- Уникальность, неупорядоченность:
 $\{13, 42\} = \{42, 13\} = \{42, 13, 42\}$
- Универсальное множество (универсум \mathcal{U}): множество всех мыслимых объектов
 - ▶ $\mathbb{N} = \{1, 2, 3, \dots\}$
 - ▶ $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
 - ▶ $\mathbb{Q} = \{m/n \mid m, n \in \mathbb{Z}; n \neq 0\}$

A является подмножеством B тогда и только тогда, когда все элементы A являются элементами B

$$A \subseteq B \iff \forall x : x \in A \Rightarrow x \in B$$

A является подмножеством B тогда и только тогда, когда все элементы A являются элементами B

$$A \subseteq B \iff \forall x : x \in A \Rightarrow x \in B$$

- $\{13, 42\} \subseteq \{7, 13, 37, 42, 99\}$
- $\{1, 3, 5, \dots\} \subseteq \mathbb{N}$
- $\mathbb{N} \subseteq \mathbb{Z}$
- $\forall A : A \subseteq A$
- Пустое множество (\emptyset): множество без элементов
 - ▶ $\forall x : x \notin \emptyset$
 - ▶ $\forall A : \emptyset \subseteq A$

Множества A и B **равны** тогда и только тогда, когда A является подмножеством B и B является подмножеством A

$$A = B \iff A \subseteq B \text{ и } B \subseteq A$$

Множества A и B **равны** тогда и только тогда, когда A является подмножеством B и B является подмножеством A

$$A = B \iff A \subseteq B \text{ и } B \subseteq A$$

A является **строгим подмножеством** B тогда и только тогда, когда A является подмножеством B , но они не равны друг другу

$$A \subset B \iff A \subseteq B \text{ и } A \neq B$$

Множества A и B **равны** тогда и только тогда, когда A является подмножеством B и B является подмножеством A

$$A = B \iff A \subseteq B \text{ и } B \subseteq A$$

A является **строгим подмножеством** B тогда и только тогда, когда A является подмножеством B , но они не равны друг другу

$$A \subset B \iff A \subseteq B \text{ и } A \neq B$$

- $\forall x : \emptyset \subset \{x\}$
- $\mathbb{N} \subset \mathbb{Z}$
- $\mathbb{Z} \not\subset \mathbb{N}$
- $\forall A : A = A \text{ и } A \not\subset A$

Множество всех подмножеств (powerset)

Множество всех подмножеств множества A состоит из всех подмножеств A

$$2^A = \{B \mid B \subseteq A\}$$

- $\forall A: \emptyset \in 2^A$
- $\forall A: A \in 2^A$
- $A = \{0, 1\} \Rightarrow \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$

Множество всех подмножеств (powerset)

Множество всех подмножеств множества A состоит из всех подмножеств A

$$2^A = \{B \mid B \subseteq A\}$$

- $\forall A: \emptyset \in 2^A$
- $\forall A: A \in 2^A$
- $A = \{0, 1\} \Rightarrow \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$

Сколько элементов может быть в множестве всех подмножеств?

Объединение: $A \cup B = \{x \mid x \in A \text{ или } x \in B\}$

Пересечение: $A \cap B = \{x \mid x \in A \text{ и } x \in B\}$

Разность: $A \setminus B = \{x \mid x \in A \text{ и } x \notin B\}$

Дополнение: $\overline{A} = \{x \mid x \in \mathcal{U} \text{ и } x \notin A\} = \mathcal{U} \setminus A$

Строки: неформально

Строка — последовательность символов

- **Алфавит** (Σ) — конечное множество (атомарных, неделимых)

СИМВОЛОВ

- ▶ $\{a, b, c, \dots, z\}$
- ▶ $\{\alpha, \beta, \gamma, \dots, \omega\}$
- ▶ $\{0, 1\}$
- ▶ $\{\underline{\text{let}}, \underline{\text{in}}, \underline{\text{where}}, \dots\}$

- **Цепочка (предложение, слово, строка)** — любая конечная последовательность символов алфавита
 - ▶ cat
 - ▶ $K\alpha T$
 - ▶ 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 α и β ($\alpha \cdot \beta = \alpha\beta$)** — результат приписывания строки β в конец строки α
 - ▶ $\forall \alpha \beta \gamma : (\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$
 - ▶ $\forall \alpha : \alpha \cdot \varepsilon = \varepsilon \cdot \alpha = \alpha$

Пример: арифметические выражения

- Алфавит $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, (,)\}$
- $1 * (2 + 3) / 4 - 5 =$
 $1 \cdot *(2 + 3) / 4 - 5 =$
 $1 * (2 + 3) \cdot / 4 - 5 =$
 $1 \cdot * \cdot (.2 \cdot + \cdot 3 \cdot) \cdot / \cdot 4 \cdot - \cdot 5 =$
 $1 * (2 + 3) / 4 - 5 \cdot \varepsilon$
- Является ли ε арифметическим выражением?

Операции над строками

- **Обращение (реверс) цепочки** a^R — цепочка, символы которой записаны в обратном порядке
 - ▶ Если $x = abc$, то $x^R = cba$
 - ▶ $\varepsilon^R = \varepsilon$
- **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
 - ▶ $|babb| = 4$
 - ▶ $|babb|_a = 1, |babb|_b = 3, |babb|_c = 0$
 - ▶ $|\varepsilon| = 0$

- Σ — алфавит
 - ▶ $\Sigma = \{0, 1\}$
- Σ^* — множество, содержащее все цепочки в алфавите Σ , включая пустую цепочку
 - ▶ $\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$
 - ▶ Сколько может быть элементов в Σ^* ?

- Σ — алфавит
 - ▶ $\Sigma = \{0, 1\}$
- Σ^* — множество, содержащее все цепочки в алфавите Σ , включая пустую цепочку
 - ▶ $\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$
 - ▶ Сколько может быть элементов в Σ^* ?
- $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$
 - ▶ $\Sigma^+ = \{0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$
 - ▶ Сколько может быть элементов в Σ^+ ?

- Σ — алфавит
 - ▶ $\Sigma = \{0, 1\}$
- Σ^* — множество, содержащее все цепочки в алфавите Σ , включая пустую цепочку
 - ▶ $\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$
 - ▶ Сколько может быть элементов в Σ^* ?
- $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$
 - ▶ $\Sigma^+ = \{0, 1, 00, 11, 01, 10, 000, 001, 011, \dots\}$
 - ▶ Сколько может быть элементов в Σ^+ ?
- *Формальный язык* в алфавите Σ — подмножество множества всех цепочек в этом алфавите.
 - ▶ Для любого языка L (в алфавите Σ) справедливо $L \subseteq \Sigma^*$
 - ▶ $L = \{0, 00, 000, \dots\} \subset \{0, 1\}^*$
 - ▶ $L = \{0, 0101, 011011011, \dots\} \subset \{0, 1\}^*$

- Язык, на котором дано описание языка
 - ▶ Естественный язык

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

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

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

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

БНФ — Бэкуса-Наура форма

- **Символ** — элементарное понятие языка
 - ▶ $+$ означает сложение в языке арифметических выражений
- **Метапеременная** — сложное понятие языка
 - ▶ Переменной $\langle \text{выражение} \rangle$ можно обозначить выражение
- **Формула**
 - ▶ $\langle \text{определяемый символ} \rangle ::= \langle \text{посл.1} \rangle \mid \dots \mid \langle \text{посл.}n \rangle$
 - ▶ В правой части формулы — альтернатива конкатенаций строк, составленных из символов и метапеременных
- **Пример: число**
 - ▶ $\langle \text{число} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{число} \rangle$
 - ▶ $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid \dots \mid 9$

Расширенная форма Бэкуса Наура (EBNF)

- Более емкие операции
- **Итерация**
 - ▶ $\langle x \rangle ::= \{ \langle y \rangle \}$ эквивалентно: $\langle x \rangle ::= \varepsilon \mid \langle y \rangle \langle x \rangle$
- **Условное вхождение**
 - ▶ $\langle x \rangle ::= [\langle y \rangle]$ эквивалентно: $\langle x \rangle ::= \varepsilon \mid \langle y \rangle$
- Скобки для группировки
 - ▶ $(\langle x \rangle \mid \langle y \rangle) \langle z \rangle$ эквивалентно: $\langle x \rangle \langle z \rangle \mid \langle y \rangle \langle z \rangle$

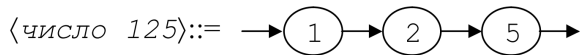
Пример: арифметические выражения

- **Терм**: последовательность цифр или любое **выражение** в скобках
- **Слагаемое**: последовательность **термов**, соединенных знаками умножения и деления
- **Выражение**: последовательность **слагаемых**, соединенных знаками сложения и вычитания (перед первым **слагаемым** может стоять минус)

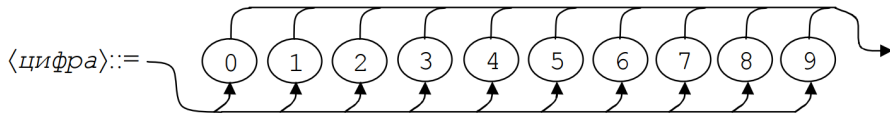
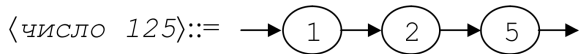
$$\begin{aligned} \langle \textit{expr} \rangle &::= [-] \langle \textit{factor} \rangle \{ (+ \mid -) \langle \textit{factor} \rangle \} \\ \langle \textit{factor} \rangle &::= \langle \textit{term} \rangle \{ (* \mid /) \langle \textit{term} \rangle \} \\ \langle \textit{term} \rangle &::= \langle \textit{number} \rangle \mid '(\langle \textit{expr} \rangle)'\end{aligned}$$

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

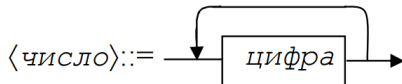
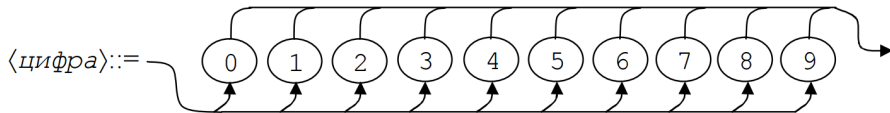
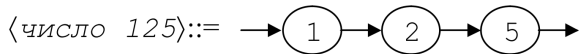
Синтаксические диаграммы Вирта



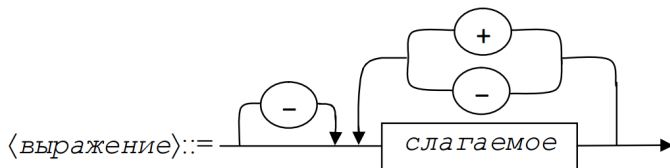
Синтаксические диаграммы Вирта



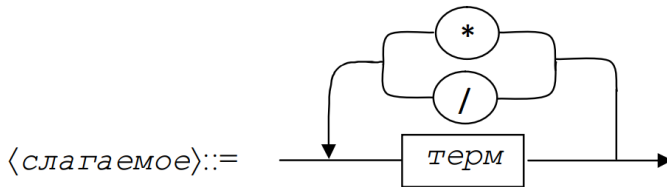
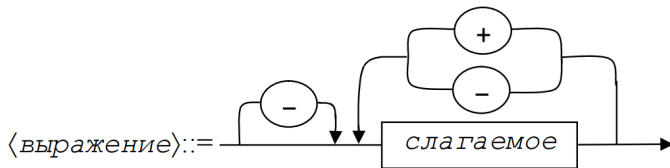
Синтаксические диаграммы Вирта



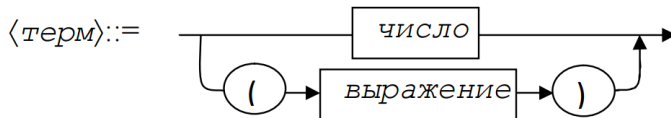
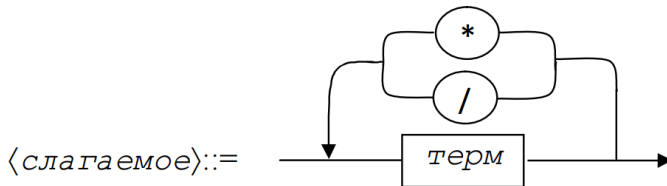
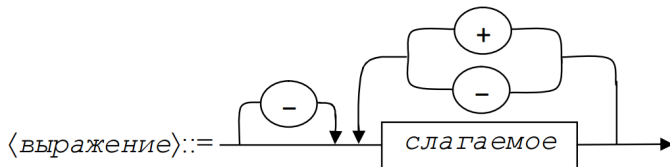
Синтаксические диаграммы Вирта



Синтаксические диаграммы Вирта



Синтаксические диаграммы Вирта



- Язык, на котором дано описание языка
 - ▶ Естественный язык
 - ▶ Язык металингвистических формул Бэкуса (БНФ)
 - ▶ Синтаксические диаграммы
 - ▶ **Граматики**
 - ▶ ...

- 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 $\alpha \rightarrow \beta$
 - ★ $\alpha \in V^* V_N V^*$
 - ★ $\beta \in V^*$
 - ▶ S — начальный нетерминал грамматики,
 - ★ $S \in V_N$

Пример: язык чисел в двоичной системе счисления

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

S	\rightarrow	0
S	\rightarrow	N
S	\rightarrow	$-N$
N	\rightarrow	$1A$
A	\rightarrow	$0A$
A	\rightarrow	$1A$
A	\rightarrow	ε

Пример: язык чисел в двоичной системе счисления

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

S	\rightarrow	0
S	\rightarrow	N
S	\rightarrow	$-N$
N	\rightarrow	$1A$
A	\rightarrow	$0A$
A	\rightarrow	$1A$
A	\rightarrow	ε

S	\rightarrow	$0 \mid N \mid -N$
N	\rightarrow	$1A$
A	\rightarrow	$0A \mid 1A \mid \varepsilon$

Пример: язык чисел в двоичной системе счисления

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

S	\rightarrow	0
S	\rightarrow	N
S	\rightarrow	$-N$
N	\rightarrow	$1A$
A	\rightarrow	$0A$
A	\rightarrow	$1A$
A	\rightarrow	ε

S	\rightarrow	$0 \mid N \mid -N$
N	\rightarrow	$1A$
A	\rightarrow	$0A \mid 1A \mid \varepsilon$

S	\rightarrow	$0 \mid [-]N$
N	\rightarrow	$1A$
A	\rightarrow	$(0 \mid 1)A \mid \varepsilon$

Отношение непосредственной выводимости

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

Отношение непосредственной выводимости: пример

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

$$S \Rightarrow -N$$

$$-N \Rightarrow -1A$$

$$-1A \Rightarrow -11A$$

Отношение выводимости является рефлексивно-транзитивным замыканием отношения непосредственной выводимости

- $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_n \in V^*$
- $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n$
- $\alpha_0 \xRightarrow{*} \alpha_n$: α_n **выводится** из α_0

Отношение выводимости: пример

$$S \rightarrow 0 \mid N \mid -N$$

$$N \rightarrow 1A$$

$$A \rightarrow 0A \mid 1A \mid \varepsilon$$

$$S \Rightarrow -N \Rightarrow -1A \Rightarrow -11A \overset{*}{\Rightarrow} -1101A \Rightarrow -1101$$

Отношение выводимости: свойства

- Транзитивность:
 $\forall \alpha, \beta, \gamma \in V^* : \alpha \xRightarrow{*} \beta, \beta \xRightarrow{*} \gamma$ следовательно $\alpha \xRightarrow{*} \gamma$
- Рефлексивность: $\forall \alpha \in V^* : \alpha \xRightarrow{*} \alpha$
- $\alpha_0 \xRightarrow{+} \alpha_n$: вывод использует хотя бы одно правило грамматики
- $\alpha_0 \xRightarrow{k} \alpha_n$: вывод происходит за k шагов

Левосторонний вывод

На каждом шагу заменяем самый **левый** нетерминал

$$\begin{aligned} S &\rightarrow AA \mid s \\ A &\rightarrow AA \mid Bb \mid a \\ B &\rightarrow c \mid d \end{aligned}$$

$$S \Rightarrow \mathbf{A}A \Rightarrow \mathbf{B}bA \Rightarrow cb\mathbf{A} \Rightarrow cb\mathbf{A}A \Rightarrow cba\mathbf{A} \Rightarrow cbaa$$

Аналогично определяется **правосторонний** вывод

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

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

Грамматика G_1 и G_2 эквивалентны, если $L(G_1) = L(G_2)$

Эквивалентность грамматик

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

$$V_T = \{0, 1, -\}$$

$$V_N = \{S, N, A\}$$

$$S \rightarrow 0 \mid N \mid -N$$

$$N \rightarrow 1A$$

$$A \rightarrow 0A \mid 1A \mid \varepsilon$$

$$V_T = \{0, 1, -\}$$

$$V_N = \{S, A\}$$

$$S \rightarrow 0 \mid 1A \mid -1A$$

$$A \rightarrow 0A \mid 1A \mid \varepsilon$$

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

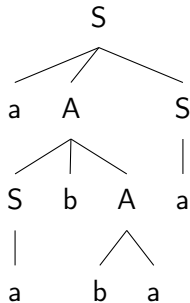
Дерево является **деревом вывода** для $G = \langle V_N, V_T, P, S \rangle$, если:

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

Пример дерева вывода

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

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



Теорема

Пусть $G = \langle V_N, V_T, P, S \rangle$ — КС-грамматика

Вывод $S \xRightarrow{*} \alpha$, где $\alpha \in V^*$, $\alpha \neq \varepsilon$ существует \Leftrightarrow существует дерево вывода в грамматике G с результатом α

Упражнение: доказать теорему