Элементы теории трансляции

Транслятор

позволяет преобразовать программу, написанную на ЯП, отличном от машинного языка, к виду, допускающему выполнение на ЭВМ.

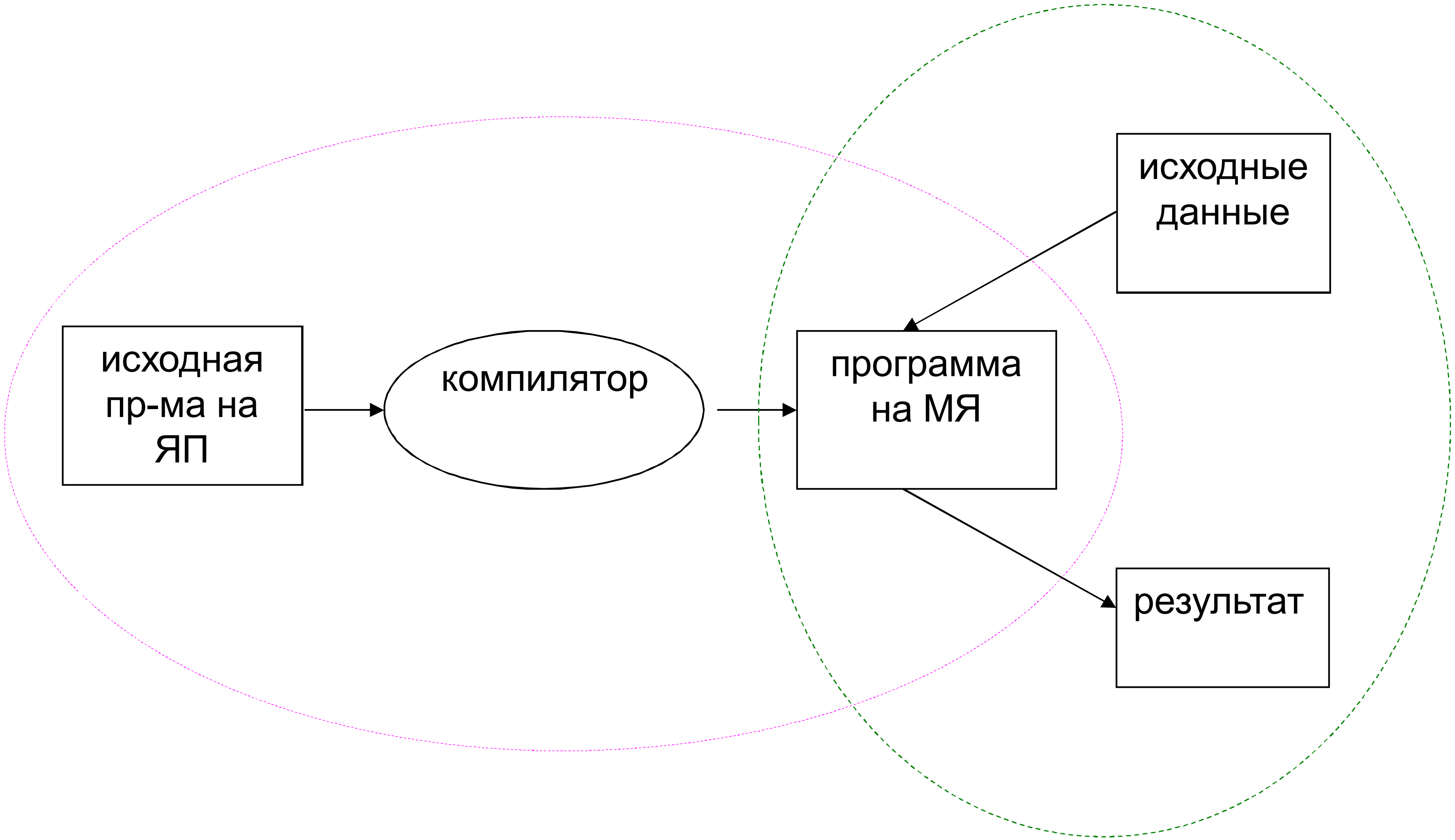
Компилятор

на вход получает программу на некотором ЯП (немашинном), а на выходе выдает объектный модуль (программу на машинном языке).

Интерпретатор

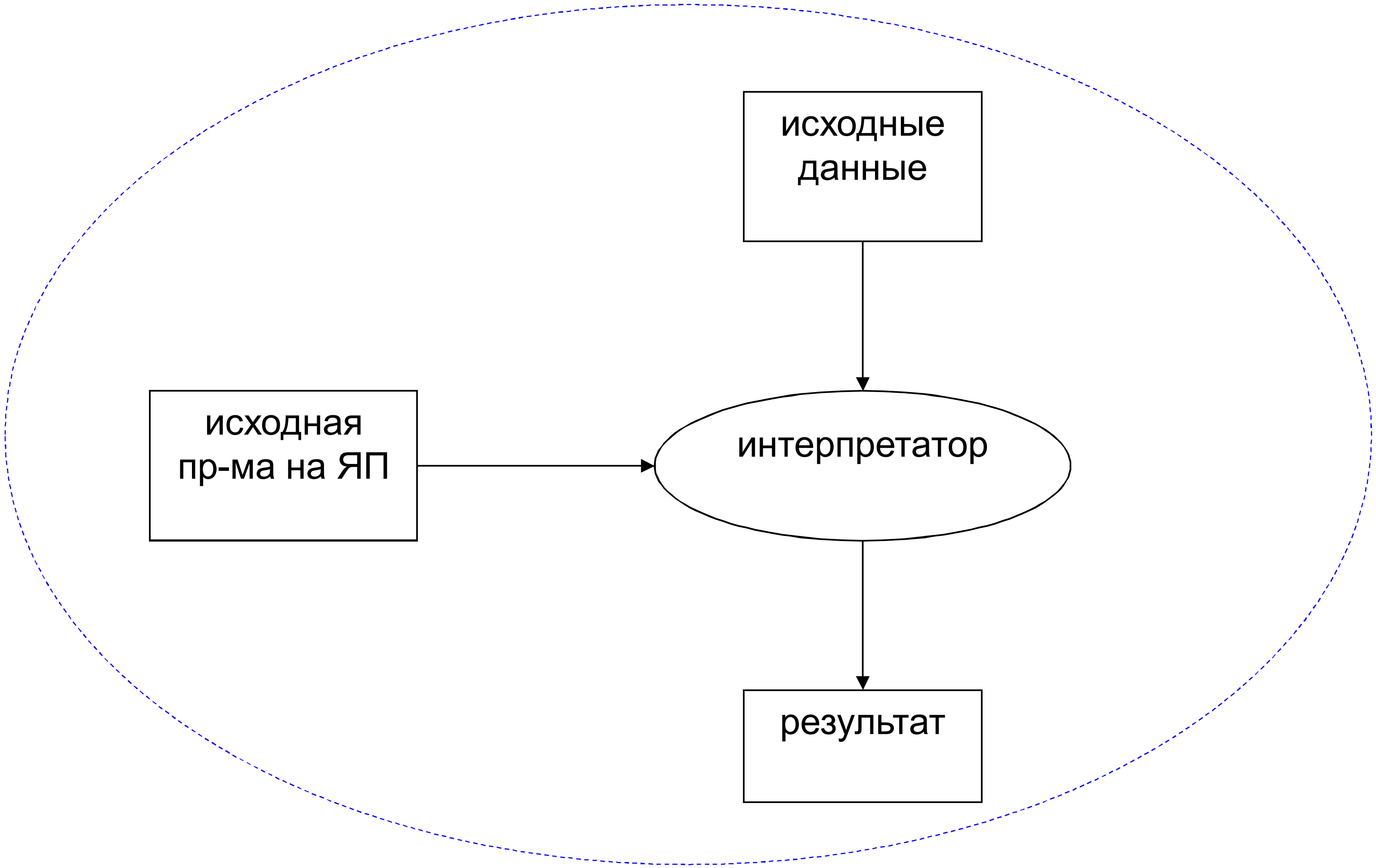
на вход получает программу на некотором ЯП (немашинном) и, считывая предложение за предложением исходной программы, анализирует их и тут же выполняет действия, указанные в этих предложениях.

Система программирования компилирующего типа



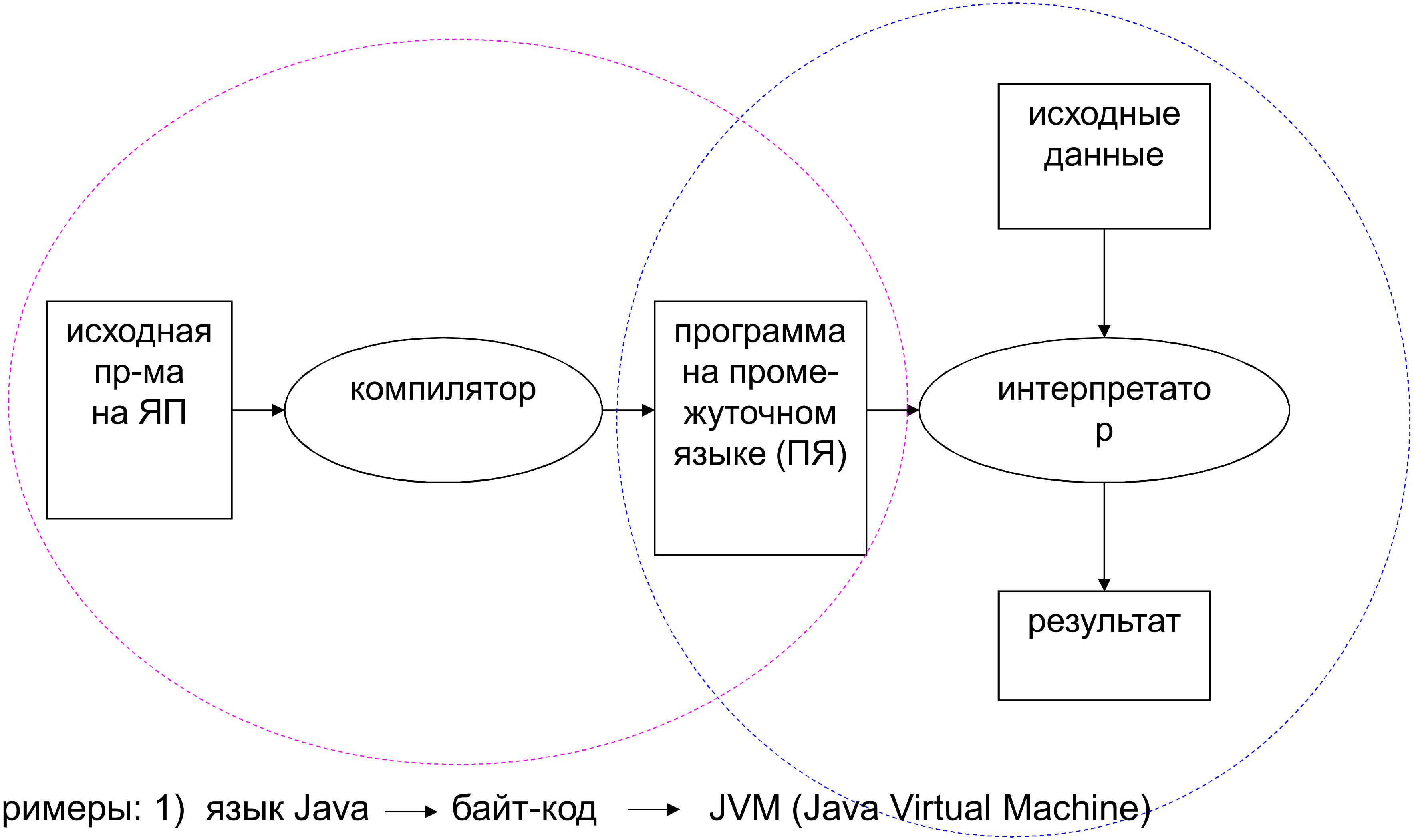
Примеры: MASM, Турбо Паскаль, gcc в UNIX

Система программирования интерпретирующего типа



Примеры: QBasic, командные интерпретаторы в UNIX -- bash, csh и др.

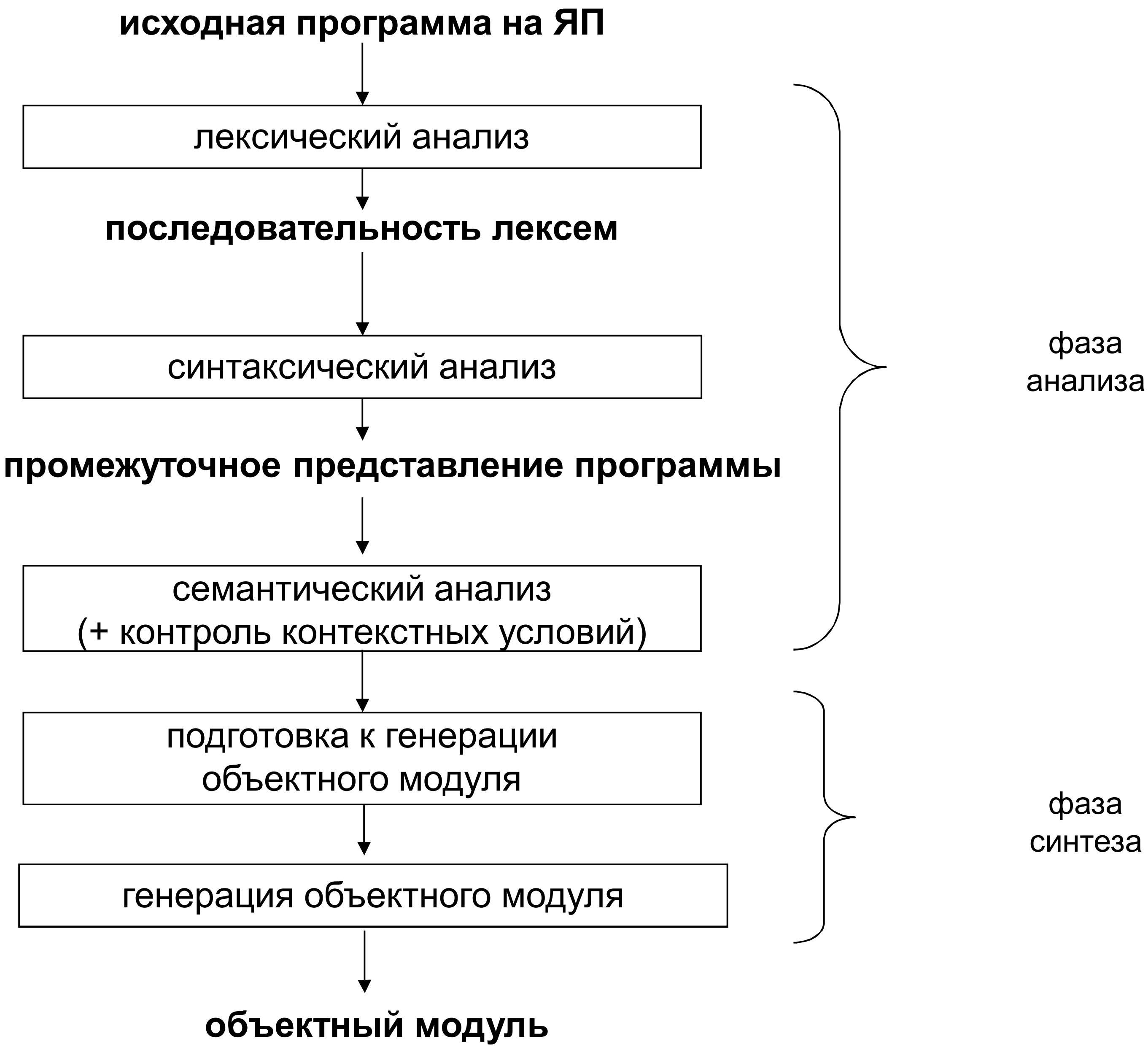
Смешанная стратегия



Примеры: 1) язык Java → байт-код → JVM (Java Virtual Machine)

2) Языки на технологии .NET (Basic,C#,C++) → промежуточный язык: CIL – Common Intermediate Language → JIT-компилятор (just-in-time)

Схема функционирования компилятора



Основные понятия теории формальных языков

Алфавит:

это конечное множество символов.

Пример: $V = \{a,b,c\}$

Цепочка символов в алфавите V :

любая конечная последовательность символов этого алфавита.

Пример: $V = \{a,b\}$. Цепочка: abbba

Пустая цепочка :

цепочка, которая не содержит ни одного символа.

Обозначение: ε

(иногда для этой цели используется символ Λ)

Конкатенация ***(сцепление)*** цепочек α и β:

цепочка, полученная приписыванием последовательности β справа к α

Обозначение: $\alpha \cdot \beta$ (или $\alpha\beta$)

Пример: если α = ab и β = cd, то

$$\alpha\beta = abcd.$$

- Для любой цепочки α верно $\alpha\varepsilon = \varepsilon\alpha = \alpha$.

  (Аналог для чисел: $\forall$ x верно $1 \cdot x = x \cdot 1 = x$)

- Операция конкатенации ассоциативна: $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$ для любых α, β, γ.

- Операция конкатенации некоммутативна:

  например, для α = ab и β = cd $\alpha \cdot \beta \neq \beta \cdot \alpha$

Обращение *(или реверс)* цепочки α:
цепочка, символы которой записаны в обратном порядке.

Обозначение: α^R

Пример: если α = abcdef, то α^R = fedcba.

Для пустой цепочки: $\varepsilon = \varepsilon^R$.

Рекурсивное определение:

$$\alpha^R = \begin{cases} \varepsilon, & \text{если } \alpha = \varepsilon; \\ \beta^R s, & \text{если } \alpha = s\beta \text{, где } s \text{ — символ алфавита} \end{cases}$$

n-я степень цепочки α (обозначается α^n) :

конкатенация n цепочек α

$$\underbrace{\alpha\ \alpha\ \alpha\ \ldots\ \alpha\ \alpha}_{n \text{ раз}} = \alpha^n$$

Рекурсивное определение:

$$\alpha^n = \begin{cases} \varepsilon, & \text{если } n=0; \\ \alpha^{n-1}\alpha, & \text{если } n>0 \end{cases}$$

Длина *цепочки* :

это число составляющих ее символов

Пример: если α = abcdefg, то длина α равна 7.

Обозначение: $|\alpha|$

$|\varepsilon| = 0$

Рекурсивное определение:

$$|\alpha| = \begin{cases} 0, & \text{если } \alpha=\varepsilon; \\ |\beta|+1, & \text{если } \alpha = \beta s, \text{ где } s \text{ — символ алфавита} \end{cases}$$

Обозначение $|\alpha|_s$ используется для числа вхождений символа s в цепочку α.

Язык в алфавите V :

подмножество цепочек конечной длины в этом алфавите.

V^* :

множество, содержащее все цепочки конечной длины в алфавите V, включая пустую цепочку ε.

Пример: $V = \{0,1\}$,

$V^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, ...\}$.

V^+ :

множество, содержащее все цепочки конечной длины в алфавите V, исключая пустую цепочку ε.

- $V^* = V^+ \cup \{\varepsilon\}$
- *Для любого языка* $L \subseteq V^*$

способы описания языков

Явное перечисление:

$L = \{ab, abb, ba, baa\}$
Язык L — конечный

Формулы:

$L = \{a^n b^n \mid n \geq 0\}$
Цепочки языка L: ε, ab, aabb, aaabbb, …

Порождающие грамматики Хомского

Распознаватели (МТ, ЛОА, конечные автоматы, МП-автоматы)

Декартово произведение множеств A и B :

множество { (a,b) | a $\in$ A, b $\in$ B} Обозначение: $A \times B$

Порождающая грамматика G :

это четверка $\langle T, N, P, S \rangle$, где

- T – алфавит *терминальных символов (терминалов)*,
- N – алфавит *нетерминальных символов (нетерминалов)*, не пересекающийся с T,
- P – конечное подмножество множества $(T \cup N)^+ \times (T \cup N)^*$;

  элемент (α, β) множества P называется *правилом вывода* и записывается в виде $\alpha \to \beta$,

  α содержит хотя бы один нетерминал;

  S – *начальный символ (цель)* грамматики, $S \in N$.

  правила

  $\alpha \to \beta_1 \quad \alpha \to \beta_2 \; \ldots \; \alpha \to \beta_n$

  записываются сокращенно

  $\alpha \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$

  β_i для $i = 1, 2, \ldots, n$, - ***альтернатива*** правила вывода из цепочки α.

Порождающая грамматика G :

это четверка $\langle T, N, P, S\rangle$, где

- T – алфавит *терминальных символов (терминалов)*,
- N – алфавит *нетерминальных символов (нетерминалов)*,

  $T \cap N = \varnothing$,
- P – конечное подмножество множества $(T \cup N)^+ \times (T \cup N)^*$;

  элемент (α, β) множества P называется *правилом вывода* и записывается в виде $\alpha \rightarrow \beta$,
- S – *начальный символ (цель)* грамматики, $S \in N$.

Пример грамматики: $G_1 = \langle \{0,1\}, \{A, S\}, P, S \rangle$

P:

$S \rightarrow 0A1$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

Пусть $\beta \in (T \cup N)^*$ $\alpha \in (T \cup N)^+$

β ***непосредственно выводима*** *из* α
в грамматике $G = \langle T, N, P, S \rangle$,
если $\alpha = \xi_1\gamma\xi_2$,
$\beta = \xi_1\delta\xi_2$,
где $\xi_1, \xi_2, \delta \in (T \cup N)^*$, $\gamma \in (T \cup N)^+$ и $\gamma \to \delta \in P$.

Обозначение: $\alpha \to \beta$

Пример: $G_1 = \langle \{0,1\}, \{A, S\}, P, S \rangle$
P:
$S \to 0A1$
$0A \to 00A1$
$A \to \varepsilon$

Цепочка 00A11 непосредственно выводима из 0A1 в грамматике G_1 : $\underline{0A}1 \to \underline{00A1}1$ (по правилу $0A \to 00A1$)

Цепочка $\beta \in (T \cup N)^*$ ***выводима*** из цепочки $\alpha \in (T \cup N)^+$
в грамматике $G = \langle T, N, P, S \rangle$,
если существуют цепочки $\gamma_0, \gamma_1, \ldots, \gamma_n$ $(n \geq 0)$, такие, что

$$\alpha = \gamma_0 \to \gamma_1 \to \ldots \to \gamma_n = \beta.$$

Обозначение: $\alpha \Rightarrow \beta$

Вывод длины n :
последовательность $\gamma_0, \gamma_1, \ldots, \gamma_n$

Любая цепочка выводится сама из себя за 0 шагов: $\alpha = \gamma_0 = \alpha$

$G_1 = \langle \{0,1\}, \{A, S\}, P, S \rangle$
P:

$$S \to 0A1$$
$$0A \to 00A1$$
$$A \to \varepsilon$$

$S \Rightarrow 000A111$ в G_1, т. к. $\exists$ вывод $S \to 0A1 \to 00A11 \to 000A111$.

Длина вывода равна 3.

Сентенциальная форма в грамматике $G = \langle T, N, P, S\rangle$:
$\alpha \in (T \cup N)^*$, для которой $S \Rightarrow \alpha$

Язык, порождаемый грамматикой $G = \langle T, N, P, S\rangle$:
множество $L(G) = \{\alpha \in T^* \mid S \Rightarrow \alpha\}$

Грамматика — это не алгоритм, а система правил подстановки, позволяющих строить выводы.

Язык, порождаемый грамматикой $G = \langle T, N, P, S \rangle$:

множество $L(G) = \{ \alpha \in T^* \mid S \Rightarrow \alpha \}$

Пример: $G_1 = \langle \{0,1\}, \{A,S\}, P, S \rangle$,

P:

(1) $S \to 0A1$

(2) $0A \to 00A1$

(3) $A \to \varepsilon$

$L(G) = ?$

$S \to_{(1)} 0A1 \to_{(3)} 01$

$S \to_{(1)} 0A1 \to_{(2)} 00A11 \to_{(3)} 0011$

$S \to_{(1)} 0A1 \to_{(2)} 00A11 \to_{(2)} 000A111 \to_{(3)} 000111$

…

Предположительно: $L_1 = \{0^n 1^n \mid n>0\}$

Язык, порождаемый грамматикой $G = \langle T, N, P, S \rangle$:

множество $L(G) = \{ \alpha \in T^* \mid S \Rightarrow \alpha \}$

Пример: $G_1 = \langle \{0,1\}, \{A,S\}, P, S \rangle$,

P:

(1) $S \rightarrow 0A1$

(2) $0A \rightarrow 00A1$

(3) $A \rightarrow \varepsilon$

Предположительно: $L_1 = \{0^n1^n \mid n>0\}$

Требуется доказать: $L(G_1) = L_1$:

(1) $L_1 \subseteq L(G_1)$, т.е. $\forall\, x \in L_1 \Rightarrow x \in L(G_1)$ (индукция по n)

(2) $L(G_1) \subseteq L_1$, т.е. $\forall\, x \in L(G_1) \Rightarrow x \in L_1$ (индукция по длине вывода)

Эквивалентность грамматик G_1 и G_2:

$L(G_1) = L(G_2)$

Пример:

$G_1 = \langle \{0,1\}, \{A,S\}, P_1, S \rangle$ и $G_2 = \langle \{0,1\}, \{S\}, P2, S \rangle$

P_1: $S \rightarrow 0A1$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

P_2: $S \rightarrow 0S1 \mid 01$

$$L(G_1) = L(G_2) = \{0^n1^n \mid n>0\}$$

Грамматики G_1 и G_2 ***почти эквивалентны***,

если $L(G_1) \cup \{\varepsilon\} = L(G_2) \cup \{\varepsilon\}$.

Пример:

$G_1 = \langle\{0,1\}, \{A,S\}, P_1, S\rangle$ и $G_2' = \langle\{0,1\}, \{S\}, P_2', S\rangle$

P_1: $S \rightarrow 0A1$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

P_2' : $S \rightarrow 0S1 \mid \varepsilon$

$L(G_1)=\{0^n1^n \mid n>0\}$, а $L(G_2') = \{0^n1^n \mid n\geq 0\}$

т.е. $L(G_2')=L(G_1) \cup \{\varepsilon\}$

$G_1 = (\{0,1\}, \{A,S\}, P_1, S)$

P_1: $S \rightarrow 0A1$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

Рассмотрим

$G_3 = (\{0,1\}, \{S\}, P, S)$, где P:

(1) $S \rightarrow 0S$

(2) $S \rightarrow 1S$

(3) $S \rightarrow \varepsilon$

Любая цепочка вида 0^n1^n порождается следующим способом:

-- n раз применить правило (1), затем

-- n раз применить правило (2)

-- и на последнем шаге применить правило (3).

Однако $L(G_3) \neq L(G_1)$,

т.к. $S \rightarrow 1S \rightarrow 10S \rightarrow 10 \in L(G_3)$,

$10 \notin L(G_1)$