

Лекция 12. Кодирование. Алфавитные коды.  
Теорема о разделимости равномерного кода.  
Теорема о разделимости префиксного кода.  
Алгоритм проверки разделимости алфавитного  
кода. Теорема Маркова.

Лектор — Селезнева Светлана Николаевна  
selezn@cs.msu.ru

факультет ВМК МГУ имени М.В. Ломоносова

Лекции на сайте <https://mk.cs.msu.ru>

# Алфавит

**Алфавитом** называется произвольное конечное множество.

Если  $A$  — алфавит, то любой элемент  $a \in A$  называется **буквой** алфавита  $A$ .

Например,  $A = \{0, 1\}$  — алфавит из двух букв: 0 и 1.

# Конечные слова в алфавите

Пусть  $A$  — алфавит.

(**Конечным**) **словом** в алфавите  $A$  назовем произвольную конечную последовательность букв из  $A$ .

Введем обозначение:  $\Lambda$  — **пустое** слово (т. е. слово без букв).

Множество всех конечных слов в алфавите  $A$  обозначим  $A^*$ .

**Длиной**  $|\alpha|$  слова  $\alpha \in A^*$  назовем число букв в нем,  $|\Lambda| = 0$ .

Если  $\alpha \in A^*$ , то  $i$ -ю букву слова  $\alpha$  обозначим  $\alpha(i)$ ,  
 $i = 1, \dots, |\alpha|$ .

Т. е. если  $|\alpha| = m$ , где  $m \geq 1$ , то  $\alpha = \alpha(1)\alpha(2) \dots \alpha(m)$ .

# Соединение слов

Если  $A$  — алфавит и  $\alpha, \beta \in A^*$ , где  $|\alpha| = k$ ,  $|\beta| = m$ , то **соединением** (или **конкатенацией**) слов  $\alpha$  и  $\beta$  назовем слово  $\alpha\beta \in A^*$ , где

$$\alpha\beta = \alpha(1) \dots \alpha(k)\beta(1) \dots \beta(m).$$

Отметим, что  $|\alpha\beta| = k + m$ .

# Подслово

Пусть  $A$  — алфавит. Слово  $\beta \in A^*$  называется **подсловом** слова  $\alpha \in A^*$ , если найдутся такие слова  $\alpha_1, \alpha_2 \in A^*$  (возможно, пустые), что

$$\alpha = \alpha_1 \beta \alpha_2.$$

# Префиксы слова

Пусть  $A$  — алфавит и  $\alpha \in A^*$ ,  $\alpha = \alpha(1)\alpha(2) \dots \alpha(m)$ .

**Префиксом** слова  $\alpha$  называется **любое его начало**, в том числе, и пустое. Т.е. **если  $\alpha = \alpha_1\alpha_2$ , где  $\alpha_1, \alpha_2 \in A^*$ , то  $\alpha_1$  — префикс слова  $\alpha$ .**

Значит, всеми возможными префиксами слова  $\alpha$  являются слова:

$$\Lambda, \alpha(1), \alpha(1)\alpha(2), \dots, \alpha(1)\alpha(2) \dots \alpha(m).$$

**Например**, если  $A = \{0, 1\}$  и  $\alpha = 0011$ , то

$$\Lambda, 0, 00, 001, 0011 —$$

все префиксы слова  $\alpha$ .

Префикс называется **собственным**, если он **не совпадает ни с пустым словом, ни со словом  $\alpha$ .**

# Суффиксы слова

Пусть  $A$  — алфавит и  $\alpha \in A^*$ ,  $\alpha = \alpha(1)\alpha(2)\dots\alpha(m)$ .

**Суффиксом** слова  $\alpha$  называется **любое его окончание**, в том числе, и пустое. Т.е. **если  $\alpha = \alpha_1\alpha_2$ , где  $\alpha_1, \alpha_2 \in A^*$ , то  $\alpha_2$  — суффикс слова  $\alpha$ .**

Значит, всеми возможными суффиксами слова  $\alpha$  являются слова:

$$\Lambda, \alpha(m), \alpha(m-1)\alpha(m), \dots, \alpha(1)\alpha(2)\dots\alpha(m).$$

Например, если  $A = \{0, 1\}$  и  $\alpha = 0011$ , то

$$\Lambda, 1, 11, 011, 0011 —$$

все префиксы слова  $\alpha$ .

Суффикс называется **собственным**, если он **не совпадает ни с пустым словом, ни со словом  $\alpha$ .**

# Кодирование

Пусть заданы два алфавита  $A$  и  $B$ .

Алфавит  $A$  назовем **исходным**, алфавит  $B$  — **кодировущим**.

**Кодированием** (из  $A$  в  $B$ ) называется произвольное отображение

$$\varphi : A^* \rightarrow B^*.$$

При кодировании  $\varphi$  любое слово  $\alpha \in A^*$  называется **сообщением**, а слово  $\beta = \varphi(\alpha) \in B^*$  — его **кодом**.

Можно рассматривать кодирования вида  $\varphi : S \rightarrow B^*$ , где  $S \subseteq A^*$  — множество (исходных) сообщений.



# Код

Если  $\varphi : A^* \rightarrow B^*$  — кодирование, то множество кодов всех слов из  $A^*$  назовем **кодом**  $C_\varphi$ , т. е.

$$C_\varphi = \{\varphi(\alpha) \mid \alpha \in A^*\} \subseteq B^*.$$

Т. е. код  $C_\varphi$  — множество кодов всех сообщений.

# Кодирования

Как правило (в приложениях), кодирование  $\varphi : A^* \rightarrow B^*$  описывается схемой (или алгоритмом) кодирования, т. е. правилом, позволяющим для любого сообщения  $\alpha \in A^*$  получить его код  $\varphi(\alpha) \in B^*$ .

Кроме того, требуется схема (или алгоритм) декодирования, т. е. правило, позволяющее для любого слова  $\beta \in B^*$  понять, является ли  $\beta$  кодом какого-то сообщения, и при положительном ответе найти такое сообщение  $\alpha \in A^*$ , что  $\varphi(\alpha) = \beta$ .

Очень желательным свойством для кодирования  $\varphi$  является **однозначность декодирования**.

# Разделимость кодирования

Кодирование  $\varphi : A^* \rightarrow B^*$  называется **однозначным** (или **разделимым**), если для любых слов  $\alpha_1, \alpha_2 \in A^*$  из  $\alpha_1 \neq \alpha_2$  следует  $\varphi(\alpha_1) \neq \varphi(\alpha_2)$ .

Т.е. кодирование  $\varphi$  — разделимо, если оно **разным** сообщениям сопоставляет **различные коды**.

Другими словами, кодирование  $\varphi$  — однозначно, если **любое** слово  $\beta \in B^*$  является кодом не более одного сообщения.

# Алфавитное кодирование

Пусть  $A = \{a_1, \dots, a_r\}$  — исходный алфавит,  
 $B = \{b_1, \dots, b_q\}$  — кодирующий алфавит.

Кодирование  $\varphi : A^* \rightarrow B^*$  называется **алфавитным** (или **побуквенным**), если оно описывается следующей схемой:

1) заданы **различные** непустые коды букв алфавита  $A$ :

$$\begin{aligned}\varphi(a_1) &= B_1, B_1 \in B^*, \\ \varphi(a_2) &= B_2, B_2 \in B^*, \\ &\dots, \\ \varphi(a_r) &= B_r, B_r \in B^*,\end{aligned}$$

2) слова в алфавите  $A$  **кодируются побуквенно**, т. е. если  $\alpha \in A^*$ ,  $\alpha = a_{i_1} a_{i_2} \dots a_{i_m}$ , где  $m \geq 2$ , то

$$\varphi(\alpha) = \varphi(a_{i_1})\varphi(a_{i_2})\dots\varphi(a_{i_m}) = B_{i_1} B_{i_2} \dots B_{i_m}.$$

# Алфавитный код

Пусть  $\varphi$  — алфавитное кодирование из  $A$  в  $B$ , т. е.

$$\varphi(a_1) = B_1, \varphi(a_2) = B_2, \dots, \varphi(a_r) = B_r.$$

Коды букв алфавита  $A$ , т. е. слова  $B_1, \dots, B_r$ , называются **кодowymi словами**.

Множество всех кодовых слов при кодировании  $\varphi$  назовем **алфавитным кодом**  $C_\varphi$ , т. е.

$$C_\varphi = \{B_1, \dots, B_r\}.$$

Отметим, что код  $C_\varphi$  однозначно определяет алфавитное кодирование  $\varphi$  (при заданном порядке букв из  $A$ ).

Алфавитный код  $C_\varphi$  назовем **однозначным** (или **разделимым**), если кодирование  $\varphi$  — **разделимо**.

# Декодирование

Пусть  $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$  — алфавитный код и  $\beta \in B^*$ .

Декодировать слово  $\beta$  означает **разбить его на последовательность кодовых слов** (если это возможно), т. е. представить в виде:

$$\beta = B_{i_1} B_{i_2} \dots B_{i_m},$$

где  $B_{i_1}, \dots, B_{i_m} \in C_\varphi$ .

Если код  $C_\varphi$  является разделимым, то для любого слова  $\beta \in B^*$  найдется **не более одного декодирования**.

# Алфавитные коды

**Пример.** Пусть  $A = \{a_1, a_2, a_3\}$ ,  $B = \{0, 1\}$  и  $\varphi_1$  — алфавитное кодирование из  $A$  в  $B$ , где

$$\varphi_1(a_1) = 00,$$

$$\varphi_2(a_2) = 01,$$

$$\varphi_3(a_3) = 11.$$

Тогда, например, для сообщения  $\alpha = a_1 a_1 a_3 \in A^*$  его код:

$$\varphi_1(\alpha) = 000011 \in B^*.$$

Код  $C_{\varphi_1}$  является разделимым (почему?).

# Равномерный алфавитный код

Алфавитный код  $C = \{B_1, \dots, B_r\} \subseteq B^*$  называется **равномерным**, если **длины всех его кодовых слов одинаковы**, т. е.

$$|B_1| = |B_2| = \dots = |B_r|.$$



# Разделимость равномерного алфавитного кода

**Предложение 12.1.** *Любой равномерный алфавитный код является разделимым.*

**Доказательство.** Пусть  $C = \{B_1, \dots, B_r\} \subseteq B^*$  — равномерный алфавитный код,  $|B_i| = l, i = 1, \dots, r$ .

Рассмотрим произвольное слово  $\beta \in B^*$ . Как можно его декодировать?

Разобьем слово  $\beta$  на последовательность слов длины  $l$ .

Если такое разбиение возможно, и каждое слово в этой последовательности является кодовым, то получили однозначное декодирование слова  $\beta$ .

Иначе, слово  $\beta$  не допускает декодирования.



# Алфавитные коды

**Пример.** Пусть  $A = \{a_1, a_2, a_3\}$ ,  $B = \{0, 1\}$  и  $\varphi_2$  — алфавитное кодирование, где

$$\begin{aligned}\varphi_2(a_1) &= 0, \\ \varphi_2(a_2) &= 10, \\ \varphi_2(a_3) &= 112.\end{aligned}$$

Тогда, например, для сообщения  $\alpha = a_1 a_1 a_3 \in A^*$  его код:

$$\varphi_2(\alpha) = 00110 \in B^*.$$

Код  $C_{\varphi_2}$  является разделимым (почему?).

# Префиксный алфавитный код

Алфавитный код  $C = \{B_1, \dots, B_r\} \subseteq B^*$  называется **префиксным**, если **никакое его кодовое слово не является префиксом никакого другого его кодового слова**, т. е.

$\nexists B_i, B_j \in C : B_i = B_j \beta_2$  для некоторого слова  $\beta_2 \in B^*$ .

# Разделимость префиксного алфавитного кода

**Предложение 12.2.** *Любой префиксный алфавитный код является разделимым.*

**Доказательство.** Пусть  $C = \{B_1, \dots, B_r\} \subseteq B^*$  — префиксный алфавитный код.

Рассмотрим произвольное слово  $\beta \in B^*$ . Как можно его декодировать?

# Разделимость префиксного алфавитного кода

**Доказательство.** Пусть  $\beta = B_{i_1}\beta_1$  для некоторого кодового слова  $B_{i_1} \in C$  и некоторого слова  $\beta_1 \in B^*$ .

Отметим, что кодовое слово  $B_{i_1}$  (если оно существует) находится **однозначно**.

Действительно, если

$$\beta = B_{i_1}\beta_1 = B_{j_1}\beta'_1,$$

где  $B_{i_1}, B_{j_1} \in C$ ,  $B_{i_1} \neq B_{j_1}$ ,  $\beta_1, \beta'_1 \in B^*$ , то **какое-то из кодовых слов  $B_{i_1}$ ,  $B_{j_1}$  является префиксом другого**, что невозможно.

Далее повторим рассуждения для слова  $\beta_1 \in B^*$ .

В итоге, либо однозначно декодируем слово  $\beta$ , либо на каком-то шаге не найдем подходящее кодовое слово, а значит, слово  $\beta$  не допускает декодирования.

# Суффиксный алфавитный код

Алфавитный код  $C = \{B_1, \dots, B_r\} \subseteq B^*$  называется **суффиксным**, если **никакое его кодовое слово не является суффиксом никакого другого его кодового слова**, т. е.

$\nexists B_i, B_j \in C : B_i = \beta_1 B_j$  для некоторого слова  $\beta_1 \in B^*$ .

**Предложение 12.3.** *Любой суффиксный алфавитный код является разделимым.*

**Доказательство** проводится подобно доказательству предыдущего утверждения, только начинать декодировать слово  $\beta$  нужно справа.

# Алфавитные коды

**Пример.** Пусть  $A = \{a_1, a_2, a_3\}$ ,  $B = \{0, 1\}$  и  $\varphi_3$  — алфавитное кодирование, где

$$\varphi_3(a_1) = 0,$$

$$\varphi_3(a_2) = 1,$$

$$\varphi_3(a_3) = 01.$$

Тогда, например, для сообщения  $\alpha = a_1 a_1 a_3 \in A^*$  его код:

$$\varphi_3(\alpha) = 0001 \in B^*.$$

Код  $C_{\varphi_3}$  не является **разделимым**, т. к. слово  $\beta = 01 \in B^*$  можно декодировать двумя способами:

$$\beta = 01,$$

$$\beta = 01,$$

т. е.  $\beta = \varphi_3(a_1 a_2) = \varphi_3(a_3)$ .

# Граф однозначности алфавитного кода

Пусть  $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$  — алфавитный код.

Построим *орграф*  $G_\varphi = (V_\varphi, E_\varphi)$  для кода  $C_\varphi$ .

1. Множество вершин  $V_\varphi$ ,  $V_\varphi \subseteq B^*$ , состоит из пустого слова  $\Lambda$  и всех тех слов в алфавите  $B$ , которые являются собственным префиксом некоторого кодового слова и одновременно собственным суффиксом некоторого кодового слова (другого или, возможно, того же) и не являются никаким кодовым словом, т. е.

$$V_\varphi = \{\Lambda\} \cup \{\beta \in B^* \mid \begin{array}{l} 1) \exists B_i \in C_\varphi : B_i = \beta\beta', \beta' \neq \Lambda; \\ 2) \exists B_j \in C_\varphi : B_j = \beta''\beta, \beta'' \neq \Lambda; \\ 3) \beta \neq B_k, k = 1, \dots, r. \end{array}\}$$



# Граф однозначности алфавитного кода

Итак,  $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$  — алфавитный код.

2. Опишем множество дуг  $E_\varphi$ : если  $\beta', \beta'' \in V_\varphi$ , то  $(\beta', \beta'') \in E_\varphi$ , если найдется такое кодовое слово  $B_i$  и такая последовательность  $D$  кодовых слов  $B_{i_1}, \dots, B_{i_k}$ , что

$$B_i = \beta' B_{i_1} \dots B_{i_k} \beta'',$$

причем если  $\beta' = \beta'' = \Lambda$ , то  $k \geq 2$ ; если  $\beta' \neq \Lambda$  или  $\beta'' \neq \Lambda$ , то  $k \geq 1$ ; если  $\beta', \beta'' \neq \Lambda$ , то  $k \geq 0$ .

При этом дуге  $(\beta', \beta'') \in E_\varphi$  приписываем пометку  $D$ , где  $D = B_{i_1}, \dots, B_{i_k}$ .

# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ .

# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .

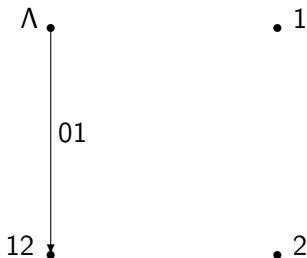
# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .

 $\Lambda \bullet$  $\bullet 1$  $12 \bullet$  $\bullet 2$

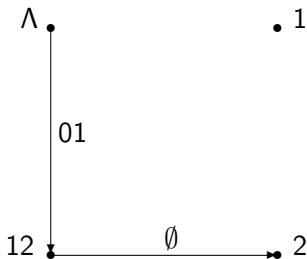
# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .



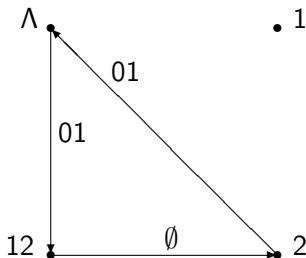
# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .



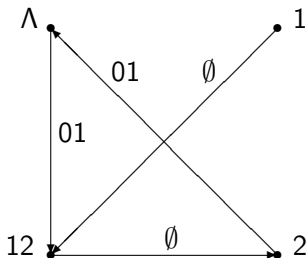
# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .



# Граф однозначности алфавитного кода

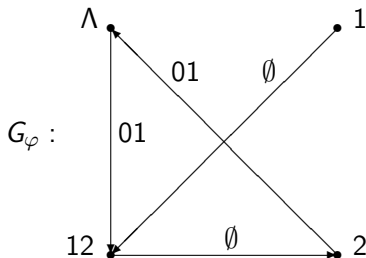
**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .





# Граф однозначности алфавитного кода

**Пример.** Пусть  $C_\varphi = \{01, 201, 112, 122, 0112\}$ . Построим граф  $G_\varphi = (V_\varphi, E_\varphi)$ . Получаем:  $V_\varphi = \{\Lambda, 1, 2, 12\}$ .



# Критерий разделимости алфавитного кода

**Теорема 12.1.** *Алфавитный код  $C_\varphi$  является разделимым тогда и только тогда, когда в графе  $G_\varphi$  отсутствуют ориентированные циклы (в том числе, и петли), проходящие через вершину  $\Lambda$ .*

**Доказательство.** Пусть  $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$  — алфавитный код и  $G_\varphi$  — граф для кода  $C_\varphi$ .

# Критерий делимости алфавитного кода

**Доказательство.** 1. Пусть код  $C_\varphi$  не является делимым.

Значит, найдется слово  $\beta \in B^*$  **наименьшей длины**, которое допускает не менее двух декодирований.

Пусть  $\beta = B'_1 B'_2 \dots B'_{t_1}$  — разбиение слова  $\beta$  на кодовые слова в 1-м декодировании и  $\beta = B''_1 B''_2 \dots B''_{t_2}$  — разбиение слова  $\beta$  на кодовые слова во 2-м декодировании.

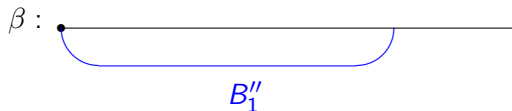
Обозначим:  $l'_i = |B'_i|$ ,  $i = 1, \dots, t_1$ , и  $l''_i = |B''_i|$ ,  $i = 1, \dots, t_2$ .

Пусть, для определенности,  $l''_1 > l'_1$ .

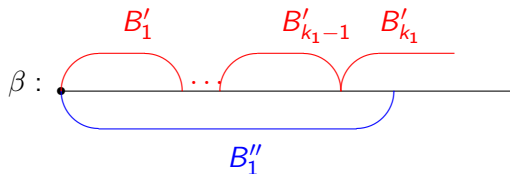
# Пояснение выбора числа $k_1$

$\beta$  : ● \_\_\_\_\_

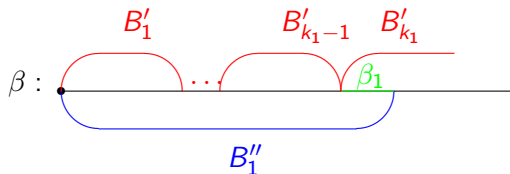
# Пояснение выбора числа $k_1$



# Пояснение выбора числа $k_1$



# Пояснение выбора числа $k_1$



# Критерий разделимости алфавитного кода

Доказательство. Найдем такое число  $k_1$ , что

$$\sum_{i=1}^{k_1-1} l'_i < l''_1, \quad \sum_{i=1}^{k_1} l'_i > l''_1.$$

Заметим, что равенства здесь быть не может, т. к. **в этом случае слово  $\beta$  можно было бы уменьшить**, что не так.

Тогда  $B''_1 = B'_1 \dots B'_{k_1-1} \beta_1$  для некоторого слова  $\beta_1 \in B^*$ ,  $\beta_1 \neq \Lambda$ .

Отметим, что слово  $\beta_1$  является собственным префиксом кодового слова  $B'_{k_1}$  и собственным суффиксом кодового слова  $B''_1$ , а также **не является никаким кодовым словом**.

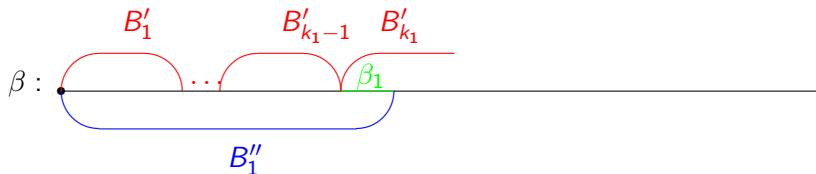
Значит, в графе  $G_\varphi$  присутствует дуга  $e_1 = (\Lambda, \beta_1) \in E_\varphi$ , которой приписана пометка  $D_1 = B'_1 \dots B'_{k_1-1}$ .



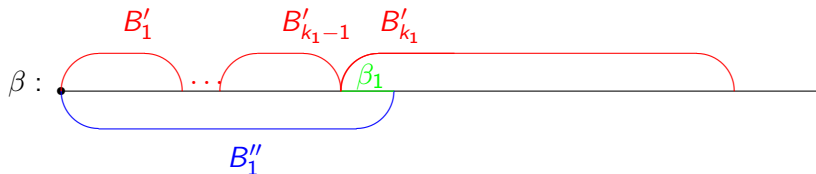
# Пояснение выбора числа $k_2$

$\beta$  : ● \_\_\_\_\_

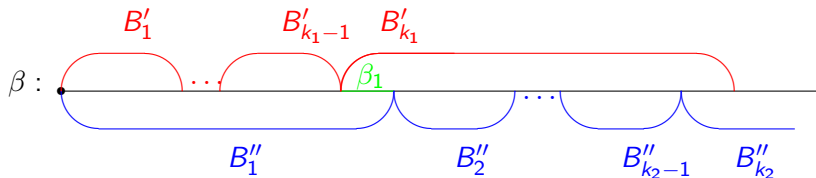
# Пояснение выбора числа $k_2$



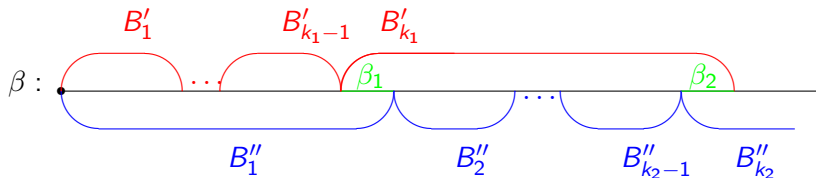
# Пояснение выбора числа $k_2$



# Пояснение выбора числа $k_2$



# Пояснение выбора числа $k_2$



# Критерий делимости алфавитного кода

Доказательство. Теперь найдем такое число  $k_2$ , что

$$|\beta_1| + \sum_{i=2}^{k_2-1} l_i'' < l_{k_1}', \quad |\beta_1| + \sum_{i=2}^{k_2} l_i'' > l_{k_1}'.$$

Снова равенства быть не может, т. к. **в этом случае слово  $\beta$  можно было бы уменьшить**, что не так.

Тогда  $B_{k_1}' = \beta_1 B_2'' \dots B_{k_2-1}'' \beta_2$  для некоторого слова  $\beta_2 \in B^*$ ,  $\beta_2 \neq \Lambda$ .

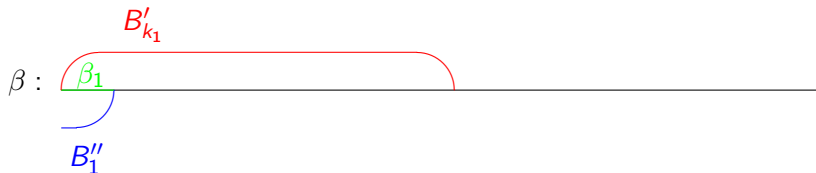
Слово  $\beta_2$  является собственным префиксом кодового слова  $B_{k_2}''$  и собственным суффиксом кодового слова  $B_{k_1}'$ , а также **не является никаким кодовым словом**.

Значит, в графе  $G_\varphi$  присутствует дуга  $e_2 = (\beta_1, \beta_2) \in E_\varphi$ , которой приписана пометка  $D_2 = B_2'' \dots B_{k_2-1}''$ .

# Пояснение выбора числа $k_3$

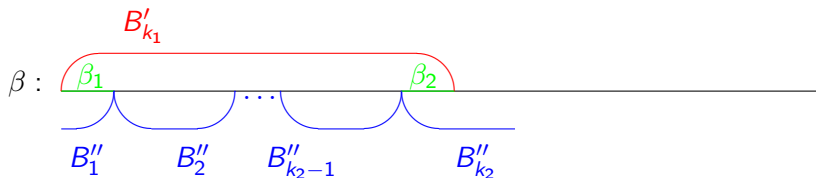
$\beta$  : \_\_\_\_\_

# Пояснение выбора числа $k_3$

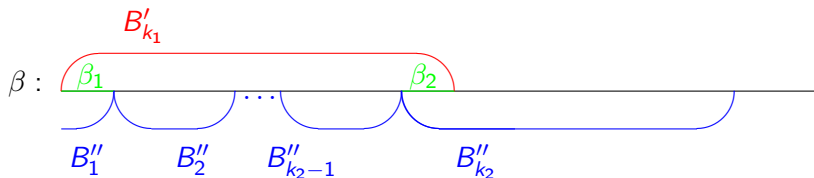




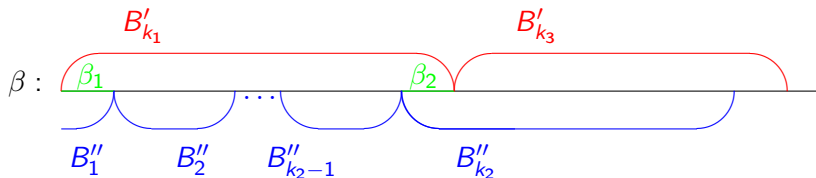
# Пояснение выбора числа $k_3$



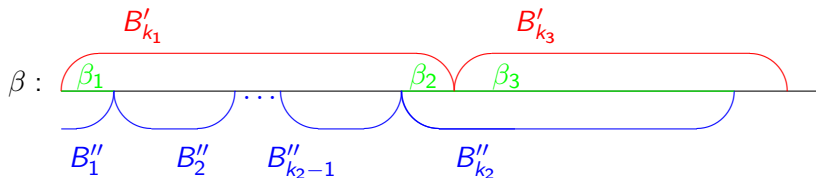
# Пояснение выбора числа $k_3$



# Пояснение выбора числа $k_3$



# Пояснение выбора числа $k_3$



# Критерий разделимости алфавитного кода

**Доказательство.** Далее найдем такое число  $k_3$ , что

$$|\beta_2| + \sum_{i=k_1+1}^{k_3-1} l'_i < l''_{k_2}, \quad |\beta_2| + \sum_{i=k_1+1}^{k_3} l'_i > l''_{k_2}.$$

Равенства быть не может, т. к. **в этом случае слово  $\beta$  можно было бы уменьшить**, что не так.

Тогда  $B''_{k_3} = \beta_2 B'_{k_1+1} \dots B'_{k_3-1} \beta_3$  для некоторого слова  $\beta_3 \in B^*$ ,  $\beta_3 \neq \Lambda$ .

Значит, в графе  $G_\varphi$  присутствует дуга  $e_3 = (\beta_2, \beta_3) \in E_\varphi$ , которой приписана пометка  $D_3 = B'_{k_1+1} \dots B'_{k_3-1}$ .

И т. д.

# Критерий делимости алфавитного кода

**Доказательство.** Через конечное число таких шагов достигнем окончания слова  $\beta$ .

Значит, в графе  $G_\varphi$  присутствует дуга  $e_{m+1} = (\beta_m, \Lambda) \in E_\varphi$  для некоторого слова  $\beta_m \in B^*$ ,  $\beta_m \neq \Lambda$ .

Этой дуге  $e_{m+1}$  приписана пометка  $D_{m+1} = B_{k_{m-1}+1}^\circ \dots B_{k_{m+1}-1}^\circ$ , где  $\circ \in \{', ''\}$  в зависимости от четности числа  $m$ .

Таким образом, в графе  $G_\varphi$  найдется ориентированный замкнутый путь:

$$P = \Lambda, e_1, \beta_1, e_2, \beta_2, \dots, \beta_m, e_{m+1}, \Lambda,$$

в котором вершина  $\Lambda$  не встречается среди вершин  $\beta_1, \dots, \beta_m$ .

Из этого пути  $P$  можно выделить **ориентированный цикл (в частности, петлю), проходящий через вершину  $\Lambda$ .**

# Критерий разделимости алфавитного кода

**Доказательство.** 2. Пусть теперь в графе  $G_\varphi$  найдется ориентированный цикл (в частности, петля)

$$P = \Lambda, e_1, \beta_1, e_2, \beta_2, \dots, \beta_m, e_{m+1}, \Lambda,$$

проходящий через вершину  $\Lambda$ .

Пусть дуге  $e_i$  приписана пометка  $D_i = B_{i_1}, \dots, B_{i_{k_i}}$ ,  
 $i = 1, \dots, m, m+1$ .

Покажем, что слово

$$\beta = D_1\beta_1 D_2\beta_2 \dots \beta_m D_{m+1} \in B^*$$

допускает не менее двух декодирований.

# Критерий разделимости алфавитного кода

Доказательство. Итак, рассмотрим слово

$$\beta = D_1\beta_1 D_2\beta_2 \dots \beta_m D_{m+1} \in B^*.$$

Пусть, для определенности,  $m$  — четно.

Первое декодирование:

$$D_1\beta_1 D_2\beta_2 D_3\beta_3 D_4 \dots D_m\beta_m D_{m+1}.$$

Второе декодирование:

$$D_1\beta_1 D_2\beta_2 D_3\beta_3 D_4\beta_4 \dots \beta_{m-1} D_m\beta_m D_{m+1}.$$

Случай нечетного  $m$  разбирается аналогично.

Значит, код  $C_\varphi$  не является разделимым.





# Проверка делимости алфавитного кода

## Алгоритм проверки делимости алфавитного кода

*Вход:* алфавитный код  $C = \{B_1, \dots, B_r\} \subseteq B^*$  в кодирующем алфавите  $B$ .

*Выход:* «да», если код  $C$  является делимым, и «нет» и слово  $\beta \in B^*$ , допускающее не менее двух декодирований, в обратном случае.

# Проверка делимости алфавитного кода

*Описание алгоритма.*

1. Построить орграф  $G$  для кода  $C$ .
2. Если граф  $G$  не содержит петель или направленных циклов, проходящих через «пустую» вершину, то выдать «да» и остановиться.
3. Иначе, пусть  $\beta_0, \beta_1, \dots, \beta_m, \beta_0$  — направленный цикл в  $G$ , где  $\beta_i \in B^*$ ,  $i = 1, \dots, m$ ,  $\beta_0 = \Lambda$ , причем дуга  $(\beta_{i-1}, \beta_i)$  помечена последовательностью  $D_i$ ,  $i = 1, \dots, m$ , а дуга  $(\beta_m, \beta_0)$  помечена последовательностью  $D_{m+1}$ . Тогда выдать «нет» и

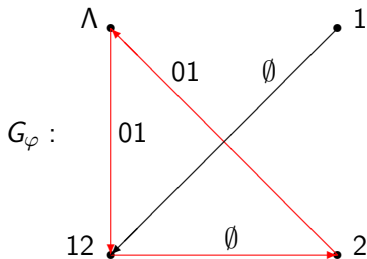
$$\beta = D_1\beta_1 D_2\beta_2 \dots \beta_m D_{m+1} \in B^*$$

и остановиться.

*Окончание описания алгоритма.*

# Проверка разделимости алфавитного кода

**Пример.** Рассмотрим код  $C_\varphi = \{01, 201, 112, 122, 0112\}$ .



Получаем:

$$\beta = 0112201 = 0112 + 201 = 01 + 122 + 01.$$

# Теорема Маркова

**Теорема 12.2 (А. А. Марков).** Пусть  $A$  — исходный алфавит,  $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$  — алфавитный код, где  $|B_i| = l_i$ ,  $i = 1, \dots, r$ . Пусть  $L = \sum_{i=1}^r l_i$  и  $w$  обозначает наибольшее число кодовых слов (возможно, с повторами), соединение которых является подсловом какого-то кодового слова. Тогда если код  $C_\varphi$  не является разделимым, то найдутся такие слова  $\alpha_1, \alpha_2 \in A^*$ ,  $\alpha_1 \neq \alpha_2$ ,  $\varphi(\alpha_1) = \varphi(\alpha_2)$ , что

$$|\alpha_1|, |\alpha_2| \leq \left\lfloor \frac{(L - r + 2)(w + 1)}{2} \right\rfloor,$$

где  $\lfloor a \rfloor$  обозначает целую часть числа  $a$ .

# Теорема Маркова

**Доказательство.** Код  $C_\varphi$  не является разделимым, значит, в графе  $G_\varphi$  найдется ориентированный цикл (в частности, петля)

$$P = \Lambda, e_1, \beta_1, e_2, \beta_2, \dots, \beta_m, e_{m+1}, \Lambda,$$

проходящий через вершину  $\Lambda$ .

Пусть дуге  $e_i$  приписана пометка  $D_i = B_{i_1}, \dots, B_{i_{k_i}}$ ,  
 $i = 1, \dots, m, m+1$ .

Можно считать, что  $P$  — петля или простой цикл, поэтому  
слова  $\beta_1, \dots, \beta_m$  — различны.

Каждое слово  $\beta_i$  является, в частности, собственным префиксом какого-то кодового слова, поэтому

$$m \leq \sum_{i=1}^r (l_i - 1) = L - r.$$

# Теорема Маркова

**Доказательство.** Рассмотрим слово

$$\beta = D_1\beta_1 D_2\beta_2 \dots \beta_m D_{m+1} \in B^*,$$

которое **допускает не менее двух декодирований**. Пусть  $\alpha_1, \alpha_2 \in A^*$ ,  $\alpha_1 \neq \alpha_2$ , — два декодирования слова  $\beta$ , т. е.

$$\beta = \varphi(\alpha_1) = \varphi(\alpha_2).$$

Слова  $\beta_1, \dots, \beta_m$  разбивают слово  $\beta$  на  $m + 1$  частей:  
 $D_1, \dots, D_{m+1}$ .

Рассмотрим  $k$  пар частей:

$$(D_1, D_2), (D_3, D_4), \dots, (D_{2k-1}, D_{2k}),$$

где  $k = \lfloor \frac{m+1}{2} \rfloor$ .

# Теорема Маркова

**Доказательство.** Для каждого  $i = 1, \dots, k$  слова

$$\begin{aligned}\beta'_i &= \overbrace{\beta_{2i-2} D_{2i-1} \beta_{2i-1}}^1 \overbrace{D_{2i}}^{\leq w}, \\ \beta''_i &= \overbrace{D_{2i-1} \beta_{2i-1} D_{2i} \beta_{2i}}^1, \\ &\quad \underbrace{\hspace{1.5cm}}_{\leq w}\end{aligned}$$

разбиваются не более, чем на  $w + 1$ , кодовых слов.

Значит, каждая пара  $(D_{2i-1}, D_{2i})$  вносит не более  $w + 1$  кодовых слов в каждое из декодирований слова  $\beta$ .

Если  $m + 1$  — нечетно, то останется еще последовательность  $D_{m+1}$ , которая также вносит не более  $w + 1$  кодовых слов в каждое из декодирований слова  $\beta$ .

# Теорема Маркова

**Доказательство.** Значит,

$$|\alpha_1|, |\alpha_2| \leq \frac{m+2}{2} \cdot (w+1) \leq \frac{(L-r+2)(w+1)}{2}.$$

Из того, что  $|\alpha_1|, |\alpha_2|$  — целые числа, получаем утверждение теоремы.





# Задачи для самостоятельного решения

1. Докажите предложение 12.3.
2. Разберите случай нечетного числа  $m$  в доказательстве теоремы 12.1.
3. Объясните, почему можно считать, что  $P$  — петля или простой цикл, в доказательстве теоремы 12.2.

# Литература к лекции

1. Алексеев В. Б. Лекции по дискретной математике. М.: Инфра-М, 2012. С. 41–47.
2. Яблонский С. В. Введение в дискретную математику. М.: Высшая школа, 2001. С. 256–272.
3. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. М.: Физматлит, 2004. Гл. VII 1.1–1.3.