

Модели вычислений

В.А. Захаров, Р.И. Подловченко

Лекция 8.

1. Дерево вывода для КС-грамматик
2. Теорема о разрастании для КС-языков
3. Примеры языков, не являющихся контекстно-свободными
4. Магазинные автоматы и их языки
5. Соответствие между КС-грамматиками и магазинными автоматами
6. Детерминированные магазинные автоматы

ДЕРЕВЬЯ ВЫВОДА ДЛЯ КС-ГРАММАТИК

Как доказать, что формальный язык **не является** контекстно-свободным?

Какие вычислительные устройства способны распознавать контекстно-свободные языки?

ДЕРЕВЬЯ ВЫВОДА ДЛЯ КС-ГРАММАТИК

Дерево вывода — это двумерный способ представления грамматического вывода для КС-грамматик.

Деревом вывода для КС-грамматики

$G = (\Sigma, \Gamma, P, S)$ называется упорядоченное корневое дерево, вершинами которого являются слово ε , терминалы и нетерминалы грамматики, удовлетворяющее следующим требованиям:

1. корнем дерева служит начальный нетерминал S ,
2. если дерево вывода содержит нетерминал N , и в грамматике есть правило $N \rightarrow x_1x_2 \dots x_k$, то вершина N имеет k вершин-потомков x_1, x_2, \dots, x_k , расположенных в указанном порядке.

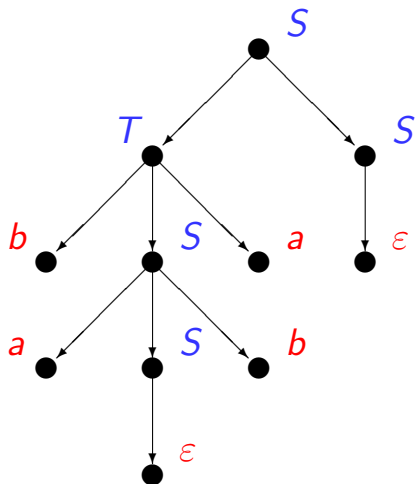
ДЕРЕВЬЯ ВЫВОДА ДЛЯ КС-ГРАММАТИК

$$S \rightarrow TS$$

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

$$T \rightarrow bSa$$



Крона дерева вывода — слово, составленное из символов на листьях дерева

$$w = baba$$

ТЕОРЕМА О РАЗРАСТАНИИ

Теорема 8.1. Пусть L — КС-язык над алфавитом Σ . Тогда существует такое положительное число N , что для любого слова w , длина $|w|$ которого превосходит N , существует такое разбиение этого слова на подслова

$$w = xuyvz,$$

что $uv \neq \varepsilon$, $|uyv| < N$, и для любого i , $i \geq 0$, верно включение

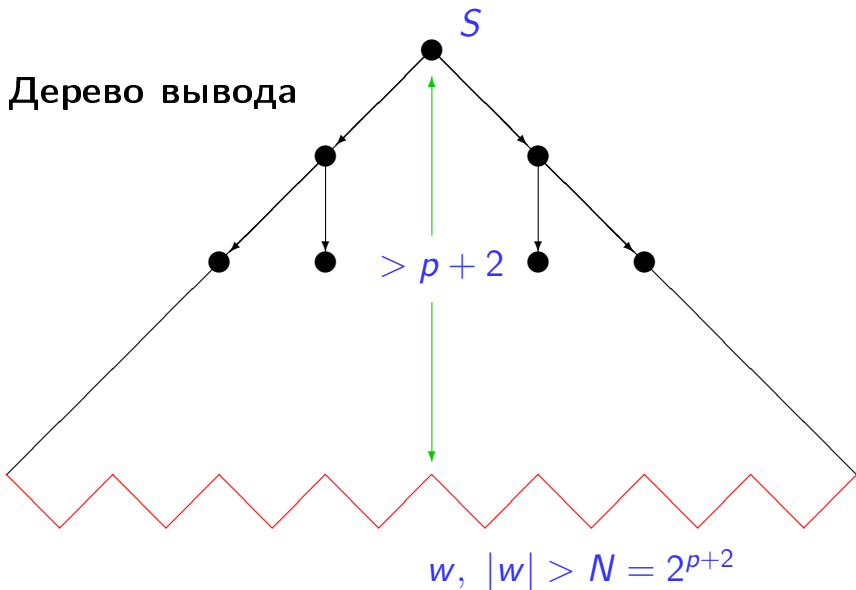
$$u^i y v^i \in L,$$

т.е. $xyz \in L$, $xuyvz \in L$, $xuuyvvz \in L$ и т.д.

ТЕОРЕМА О РАЗРАСТАНИИ

Доказательство. Если L — КС-язык, то существует КС-грамматика в нормальной форме Хомского G , порождающая этот язык. Все правила грамматики G имеют вид $S \rightarrow \varepsilon$, $N \rightarrow a$, $N \rightarrow N'N''$, и поэтому любое дерево вывода — это бинарное дерево; ветвление может отсутствовать только у вершин, потомки которых — терминалы. Обозначим символом p количество нетерминалов грамматики G . Положим $N = 2^{p+2}$. Тогда для любого слова w языка L , имеющего длину $|w| > N$, дерево его вывода для КС-грамматики G имеет высоту, превышающую величину $p + 2$.

ТЕОРЕМА О РАЗРАСТАНИИ



ТЕОРЕМА О РАЗРАСТАНИИ

Значит, в этом дереве вывода есть хотя бы одна ветвь

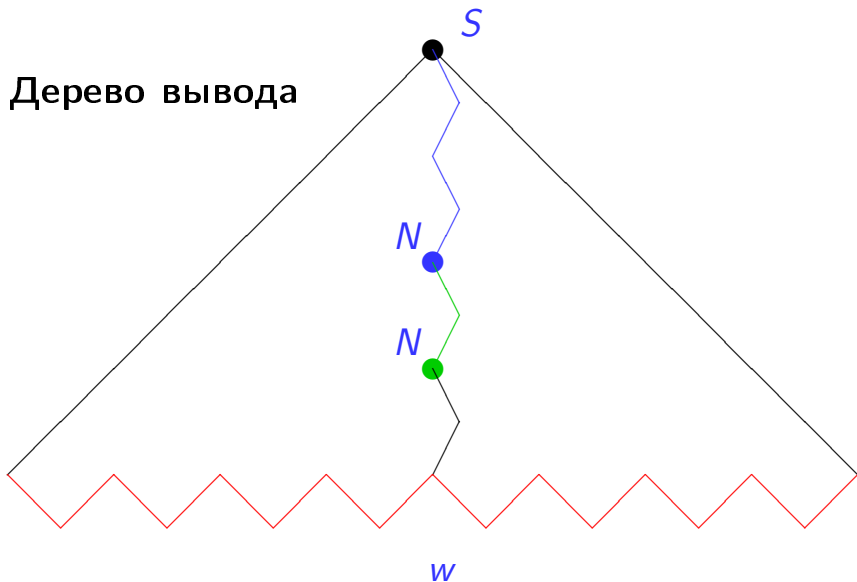
$$S \longrightarrow N_1 \longrightarrow N_2 \longrightarrow \cdots \longrightarrow N_p \longrightarrow a,$$

проходящая не менее чем через $p + 1$ вершин-нетерминалов.

Возьмем самую длинную из таких ветвей. Хотя бы один нетерминал встречается в ней дважды.

Возьмем ту пару повторно встречающихся нетерминалов N , которые расположены наиболее близко к нижнему концу этой ветви.

ТЕОРЕМА О РАЗРАСТАНИИ

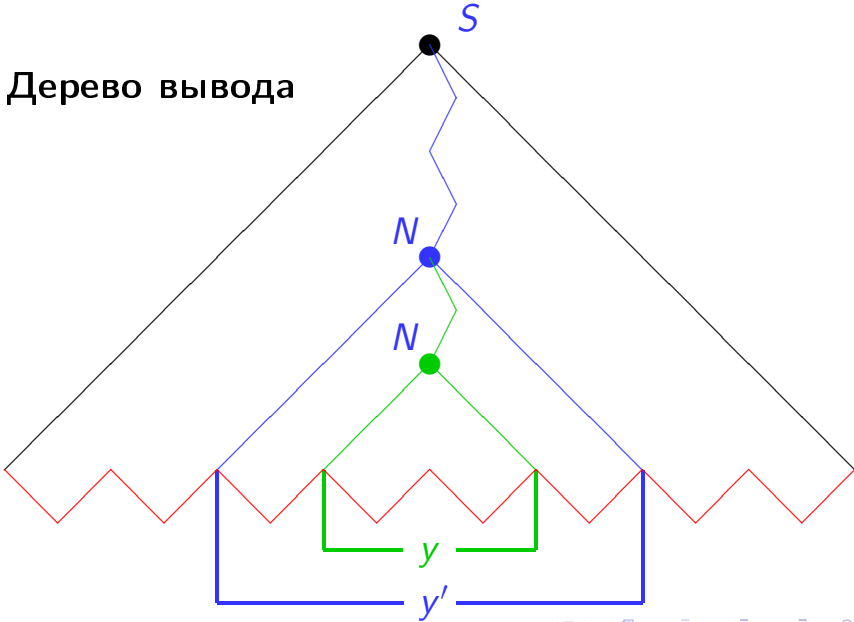


ТЕОРЕМА О РАЗРАСТАНИИ

Рассмотрим те терминальные подслова y и y' слова w , которые выводимы в нашем дереве из этих двух самых нижних повторных вхождений одного и того же нетерминала N .

ТЕОРЕМА О РАЗРАСТАНИИ

Дерево вывода



ТЕОРЕМА О РАЗРАСТАНИИ

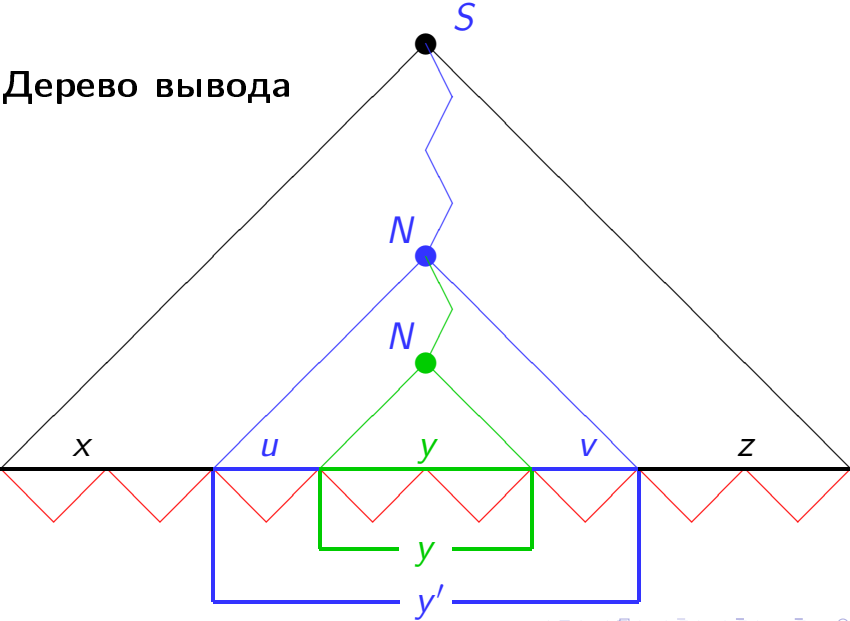
Тогда терминальное слово y' имеет разбиение $y' = uvv$, причем $uv \neq \emptyset$ (ведь нормальная форма Хомского не допускает ε -правил для не-начальных терминалов, и поэтому $y \neq y'$).

Кроме того, верно, что $|y'| < N$, поскольку выбраны были самые нижние повторяющиеся терминалы. За счет этого высота дерева вывода слова y' не превосходит $p + 2$.

А само терминальное слово w имеет разбиение $w = xy'z$.

ТЕОРЕМА О РАЗРАСТАНИИ

Дерево вывода



ТЕОРЕМА О РАЗРАСТАНИИ

Полученное разбиение свидетельствует о том, что в грамматике G выводимы следующие строки:

$$S \xrightarrow{*}_G xLz, S \xrightarrow{*}_G xuLvz, L \xrightarrow{*}_G uLv, L \xrightarrow{*}_G y.$$

И это означает, что в грамматике G выводимы следующие слова:

$$S \xrightarrow{*}_G xLz \xrightarrow{*}_G xyz,$$

$$S \xrightarrow{*}_G xLz \xrightarrow{*}_G \cdots \xrightarrow{*}_G x \underbrace{u \dots u}_i y \underbrace{v \dots v}_i z.$$

ТЕОРЕМА О РАЗРАСТАНИИ

Теорема о разрастании утверждает, что если слово из КС-языка «достаточно» длинное,

$$w \in L$$

ТЕОРЕМА О РАЗРАСТАНИИ

Теорема о разрастании утверждает, что если слово из КС-языка «достаточно» длинное,

$$\overline{w \in L}$$

то в этом слове есть такой «короткий» фрагмент,

$$\overline{u y v}$$
$$w \in L$$

ТЕОРЕМА О РАЗРАСТАНИИ

Теорема о разрастании утверждает, что если слово из КС-языка «достаточно» длинное,

$$\overline{w \in L}$$

то в этом слове есть такой «короткий» фрагмент,

$$\overline{u y v}$$
$$w \in L$$

который можно «раздвигать» в обе стороны

$$\overline{u u u u y v v v v}$$
$$w''' \in L$$

ТЕОРЕМА О РАЗРАСТАНИИ

Теорема о разрастании — это необходимое условие того, что язык является контекстно-свободным.

Это условие можно использовать для доказательства того, что некоторые языки не являются контекстно-свободными.

Пример. Язык $L = \{a^n b^n c^n : n \geq 0\}$ не является контекстно-свободным.

Доказательство проводится по стереотипной схеме. Предположим противное. Тогда существует такое N , что все слова $a^n b^n c^n$, где $n > N$ должны удовлетворять теореме о разрастании.

ТЕОРЕМА О РАЗРАСТАНИИ

Значит, «достаточно» длинное слово $a^n b^n c^n$

a^n b^n c^n

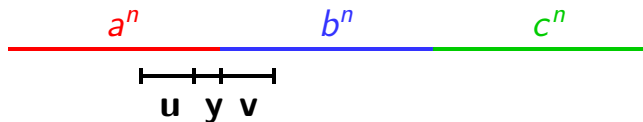


ТЕОРЕМА О РАЗРАСТАНИИ

Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.

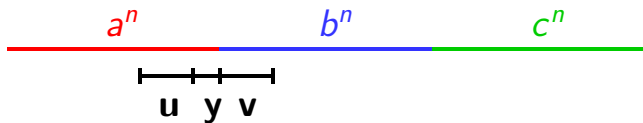


ТЕОРЕМА О РАЗРАСТАНИИ

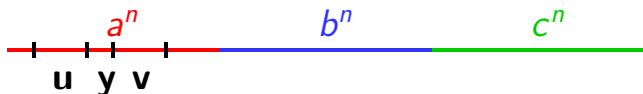
Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.



Где же расположен этот фрагмент?



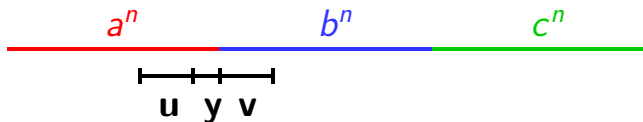
Здесь?

ТЕОРЕМА О РАЗРАСТАНИИ

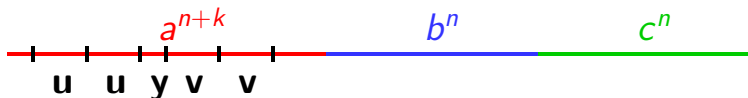
Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.



Где же расположен этот фрагмент?



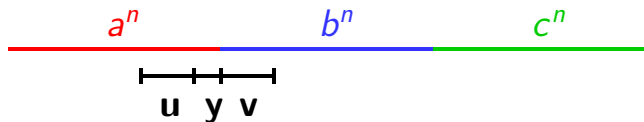
НЕТ!

ТЕОРЕМА О РАЗРАСТАНИИ

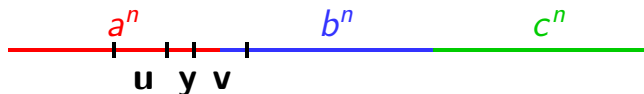
Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.



Где же расположен этот фрагмент?



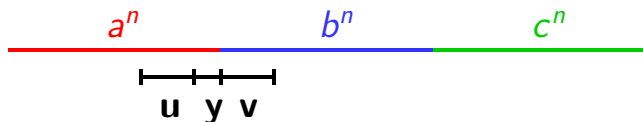
Здесь?

ТЕОРЕМА О РАЗРАСТАНИИ

Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.



Где же расположен этот фрагмент?



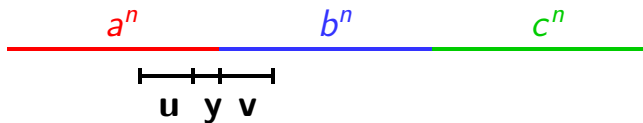
НЕТ!

ТЕОРЕМА О РАЗРАСТАНИИ

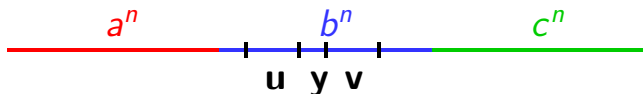
Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.



Где же расположен этот фрагмент?



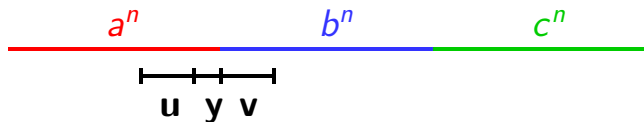
Здесь?

ТЕОРЕМА О РАЗРАСТАНИИ

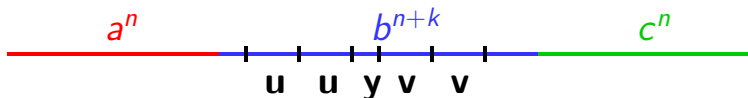
Значит, «достаточно» длинное слово $a^n b^n c^n$



должно содержать «короткий» фрагмент, способный «разрастаться», оставляя слово в языке.



Где же расположен этот фрагмент?



НЕТ!

ТЕОРЕМА О РАЗРАСТАНИИ

Где бы мы не выделяли такой «короткий» фрагмент, его «разрастание» выводит слово $a^n b^n c^n$ за пределы языка L .

Возникает противоречие с утверждением Теоремы 8.1 (о разрастании).

Источник противоречия — предположение о том, что язык L является контекстно-свободным.

Значит, язык $L = \{a^n b^n c^n : n \geq 0\}$ не является контекстно-свободным. QED

ТЕОРЕМА О РАЗРАСТАНИИ

Задача 1. Докажите, что язык

$$\bar{L} = \Sigma^* \setminus \{a^n b^n c^n : n \geq 0\}$$

является контекстно-свободным.

Задача 2. Докажите, что язык

$$L' = \{ww : w \in \Sigma^*\}$$

не является контекстно-свободным.

Задача 3. Является ли контекстно-свободным язык

$$\bar{L}' = \Sigma^* \setminus \{ww : w \in \Sigma^*\}?$$

МАГАЗИННЫЕ АВТОМАТЫ

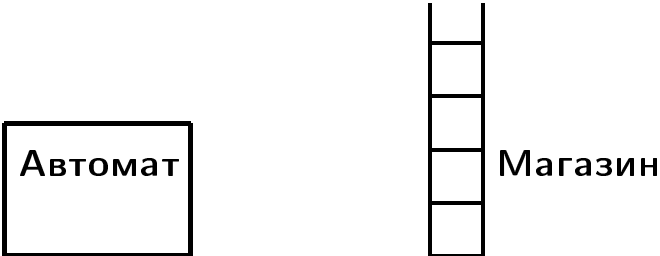
А какие вычислительные устройства способны распознавать контекстно-свободные языки?

Вычислительных возможностей конечных автоматов для этого недостаточно; их нужно усилить.

Оказывается, что для этого достаточно снабдить конечный автомат стековой (магазинной) памятью.

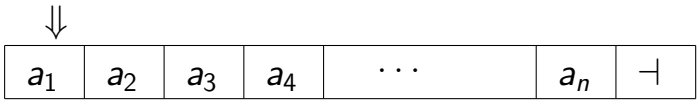
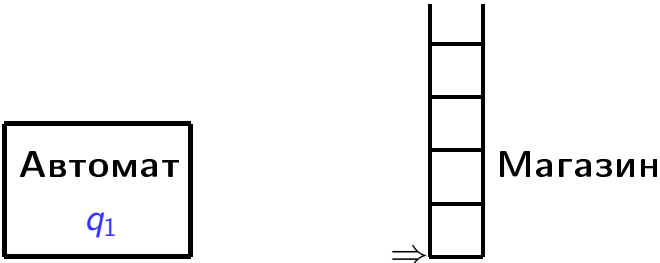
Так появилась модель вычислений магазинного автомата.

МАГАЗИННЫЕ АВТОМАТЫ



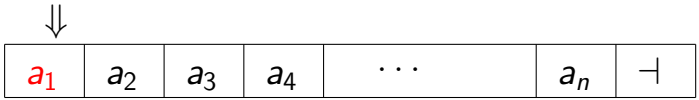
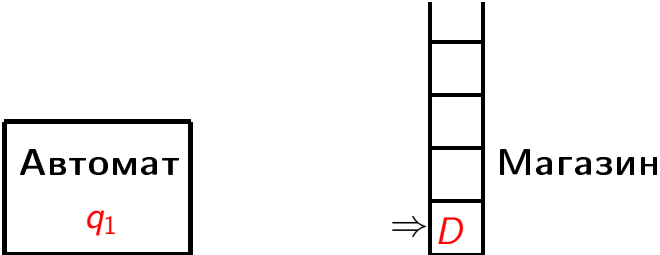
Конечный автомат снабжен дополнительной памятью — магазином (стеком)

МАГАЗИННЫЕ АВТОМАТЫ



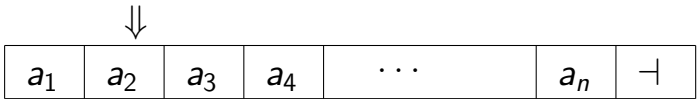
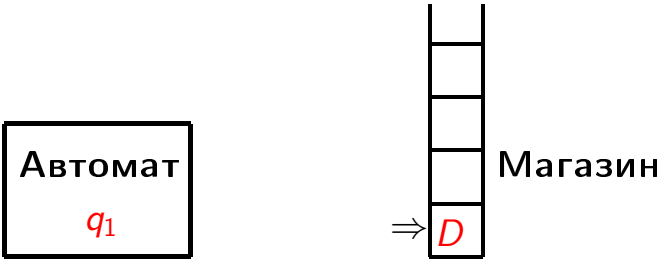
В начале работы автомат находится в начальном состоянии, а магазин пуст

МАГАЗИННЫЕ АВТОМАТЫ



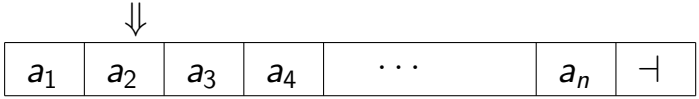
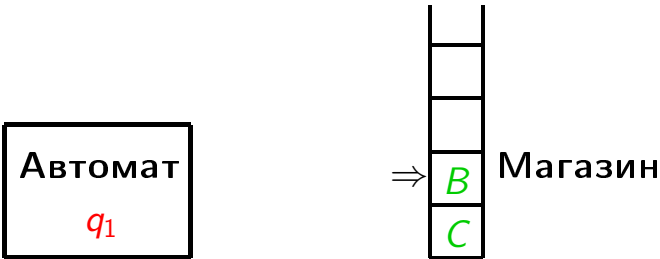
В зависимости от состояния автомата, обозреваемой буквы и символа на верхушке магазина...

МАГАЗИННЫЕ АВТОМАТЫ



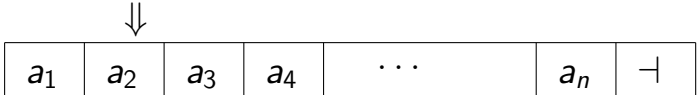
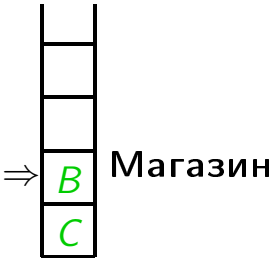
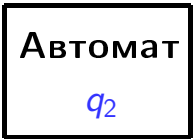
автомат может перейти к прочтению следующей буквы на ленте,...

МАГАЗИННЫЕ АВТОМАТЫ



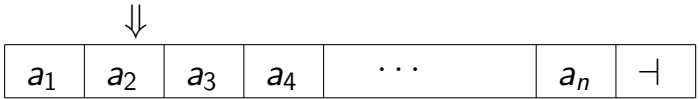
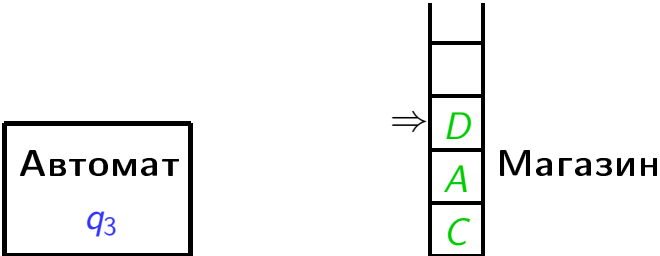
прочитать и стереть символ на верхушке магазина
и записать несколько магазинных символов,...

МАГАЗИННЫЕ АВТОМАТЫ



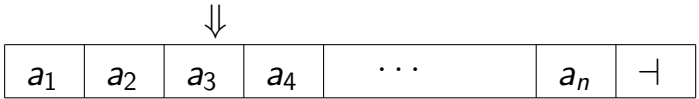
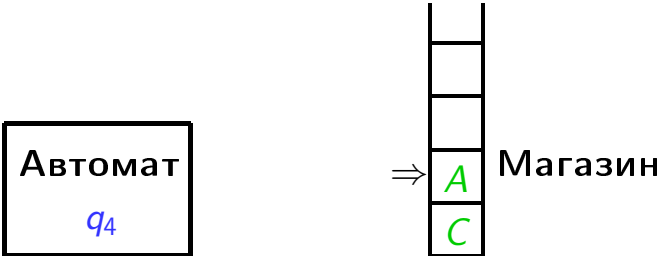
и перейти в новое состояние.

МАГАЗИННЫЕ АВТОМАТЫ



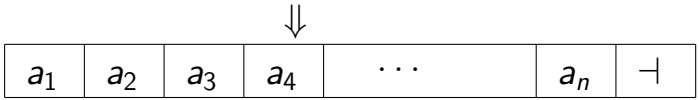
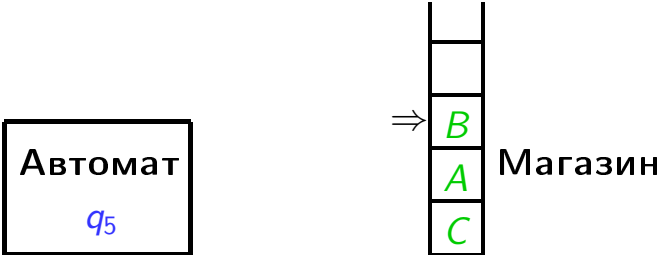
Автомат может не прочитывать очередную букву (совершает ϵ -переход.)

МАГАЗИННЫЕ АВТОМАТЫ



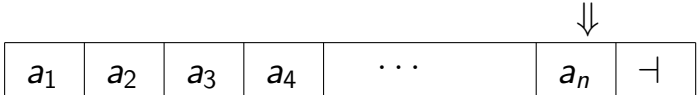
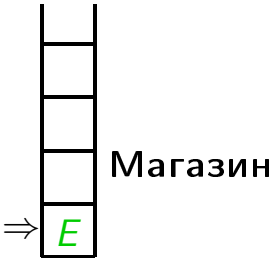
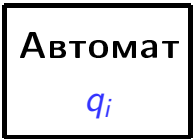
Автомат может не записывать новых символов в магазин (стирает прочитанный символ).

МАГАЗИННЫЕ АВТОМАТЫ



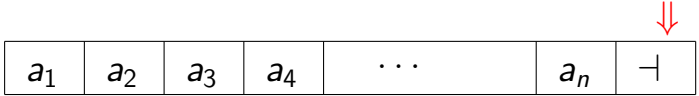
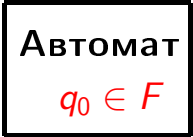
Автомат может не прочитывать и не стирать верхний символ в магазине, а записать новые символы.

МАГАЗИННЫЕ АВТОМАТЫ



Если после прочтения последней буквы слова...

МАГАЗИННЫЕ АВТОМАТЫ



автомат оказывается в допускающем состоянии и магазин пуст, то считается, что автомат успешно распознал ленточное слово

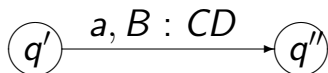
МАГАЗИННЫЕ АВТОМАТЫ

Автоматом с магазинной памятью (магазинным автоматом, стековым автоматом) называется система $\mathcal{B} = (\Sigma, \Gamma, Q, I, F, T)$, в которой

- ▶ Σ — конечный ленточный алфавит,
- ▶ Γ — конечный магазинный алфавит,
- ▶ Q — конечное множество состояний,
- ▶ I — подмножество начальных состояний,
 $I \subseteq S$,
- ▶ F — подмножество финальных состояний,
 $F \subseteq S$,
- ▶ $T \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times Q \times \Gamma^*$ — отношение переходов.

МАГАЗИННЫЕ АВТОМАТЫ

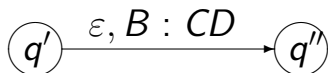
Переходы магазинного автомата можно изображать на диаграммах так:



Прочитывает на ленте букву a ,
прочитывает на вершшке магазина символ B ,
стирает его и записывает CD .

МАГАЗИННЫЕ АВТОМАТЫ

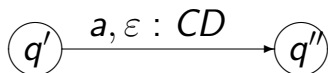
Переходы магазинного автомата можно изображать на диаграммах так:



Не читает букв на ленте,
прочитывает на вершшке магазина символ B ,
стирает его и записывает CD .

МАГАЗИННЫЕ АВТОМАТЫ

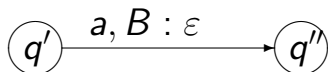
Переходы магазинного автомата можно изображать на диаграммах так:



Прочитывает на ленте букву a ,
не обращает внимания на вершущку магазина,
и записывает над ней CD .

МАГАЗИННЫЕ АВТОМАТЫ

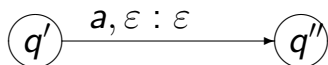
Переходы магазинного автомата можно изображать на диаграммах так:



Прочитывает на ленте букву a ,
прочитывает на вершшке магазина символ B ,
и стирает его, ничего не записывая.

МАГАЗИННЫЕ АВТОМАТЫ

Переходы магазинного автомата можно изображать на диаграммах так:

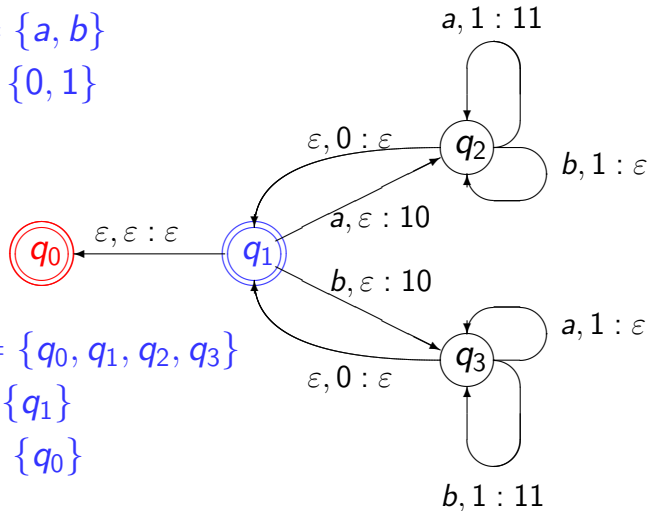


Прочитывает на ленте букву a ,
не обращает внимания на магазин,
и оставляет его без изменения.

МАГАЗИННЫЕ АВТОМАТЫ

$$\Sigma = \{a, b\}$$

$$\Gamma = \{0, 1\}$$



$$Q = \{q_0, q_1, q_2, q_3\}$$

$$I = \{q_1\}$$

$$F = \{q_0\}$$

МАГАЗИННЫЕ АВТОМАТЫ

Конфигурацией магазинного автомата

$\mathcal{B} = (\Sigma, \Gamma, Q, I, F, T)$ называется всякая пара (q, α) , где $q \in Q$ (состояние, в котором находится автомат) и $\alpha \in \Gamma^*$ (содержимое магазина).

Отношение переходов T можно распространить на множество конфигураций магазинного автомата: для любой буквы или пустого слова x , пары состояний q, q' , магазинного символа или пустой строки Y и строки магазинных символов α

$$(q, Y\alpha) \xrightarrow{x} (q', \beta\alpha) \iff (q, x, A, q', \beta).$$

МАГАЗИННЫЕ АВТОМАТЫ

Вычислением магазинного автомата \mathcal{B}

называется всякая конечная последовательность конфигураций

$$conf_0 \xrightarrow{x_1} conf_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} conf_n.$$

Говорят, что это вычисление прочитывает слово $w = x_1x_2 \dots x_n$; для обозначения такого вычисления используется запись $conf_0 \xrightarrow{w}_* conf_n$.

Вычисление называется

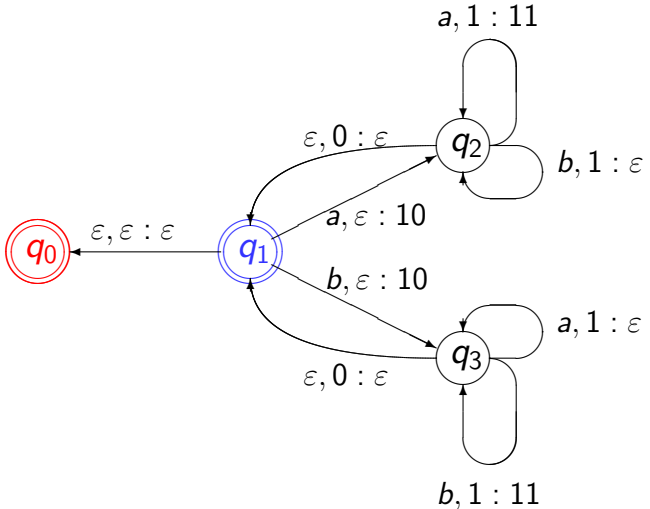
- ▶ **начальным**, если $conf_0 = (q_1, \varepsilon)$, где $q_1 \in I$;
- ▶ **финальным**, если $conf_n = (q_0, \varepsilon)$, где $q_0 \in F$;
- ▶ **успешным**, если оно является начальным и финальным.

МАГАЗИННЫЕ АВТОМАТЫ

Язык магазинного автомата \mathcal{B} — это множество $L(\mathcal{B})$ всех слов w , для которых существует успешное вычисление $conf_0 \xrightarrow{w}_* conf_n$.

Два магазинных автомата \mathcal{B}_1 и \mathcal{B}_2 называются эквивалентными, если $L(\mathcal{B}_1) = L(\mathcal{B}_2)$

МАГАЗИННЫЕ АВТОМАТЫ



$L(\mathcal{B}) = \{w : w \in \{a, b\}^*, \text{ кратности вхождения } a \text{ и } b \text{ равны}\}$

МАГАЗИННЫЕ АВТОМАТЫ

Теорема 8.2. Любой КС-язык распознается некоторым магазинным автоматом.

Доказательство. Возьмем произвольный КС-язык. Согласно теореме 7.18 этот язык порождается грамматикой в нормальной форме Хомского $G = (\Sigma, \mathcal{N}, P, S)$. Этой грамматике соответствует магазинный автомат $B_G = (\Sigma, \mathcal{N}, \{q_1, q_0\}, \{q_1\}, \{q_0\}, T)$, в котором отношение переходов определяется следующими соотношениями:

1. в T содержится переход $(q_1, \varepsilon, \varepsilon, q_0, S) \in T$,
2. если $S \rightarrow \varepsilon \in P$, то $(q_0, \varepsilon, S, q_0, \varepsilon) \in T$,
3. если $A \rightarrow BC \in P$, то $(q_0, \varepsilon, A, q_0, BC) \in T$,

МАГАЗИННЫЕ АВТОМАТЫ

Индукцией по длине грамматического вывода (или вычисления автомата) можно показать, что успешное вычисление магазинного автомата B_G

$$(q_1, \varepsilon) \xrightarrow{\varepsilon} (q_0, S) \xrightarrow{x_1} (q_0, \alpha_1) \xrightarrow{x_2} \dots \xrightarrow{x_n} (q_0, \varepsilon),$$

прочитывающее слово $w = x_1x_2 \dots x_n$ существует тогда и только тогда, когда существует успешный левосторонний вывод

$$S \xrightarrow{G} x_1\alpha_1 \xrightarrow{G} x_1x_2\alpha_2 \xrightarrow{G} \dots \xrightarrow{G} x_1x_2x_3 \dots x_n,$$

этого же слова в грамматике G .

QED

МАГАЗИННЫЕ АВТОМАТЫ

Утверждение 8.3. Для любого магазинного автомата $\mathcal{B} = (\Sigma, \Gamma, Q, I, F, T)$ существует эквивалентный ему магазинный автомат $\mathcal{B}' = (\Sigma, \Gamma, Q', I, F, T')$, у которого $|I| = |F| = 1$, и для любого перехода $(q', x, Y, q'', \alpha) \in T'$ выполняется неравенство $|Y| + |\alpha| \leq 1$.

Доказательство. Любой переход $(q', y, A, q'', B_1 B_2 \dots B_k) \in T$ можно заменить множеством переходов

$$(q', y, A, q_1, \varepsilon), (q_1, \varepsilon, \varepsilon, q_2, B_1), \dots, (q_k, \varepsilon, \varepsilon, q'', B_k)$$

А как добиться выполнения требования $|I| = |F| = 1$ придумайте самостоятельно. QED

МАГАЗИННЫЕ АВТОМАТЫ

Магазинные автоматы обладают свойством монотонности по отношению к магазину.

Утверждение 8.4. Для любого магазинного автомата $\mathcal{B} = (\Sigma, \Gamma, Q, I, F, T)$ если существует вычисление

$$(p, \alpha) \xrightarrow{w}^* (q, \beta),$$

то для любой строки магазинных символов γ существует вычисление

$$(p, \alpha\gamma) \xrightarrow{w}^* (q, \beta\gamma).$$

Доказательство. Самостоятельно.

МАГАЗИННЫЕ АВТОМАТЫ

Теорема 8.5. Любой магазинный автомат распознает КС-язык.

Доказательство. Согласно Утверждению 8.3 достаточно ограничиться рассмотрением только магазинных автоматов $\mathcal{B} = (\Sigma, \Gamma, Q, q_1, q_0, T)$, в которых каждый переход (q', x, Y, q'', α) удовлетворяет неравенству $|Y| + |\alpha| \leq 1$.

Соответствующая этому автомату КС-грамматика $G_{\mathcal{B}} = (\Sigma, \mathcal{N}, P, S)$ устроена так.

МАГАЗИННЫЕ АВТОМАТЫ

Множество нетерминалов $\mathcal{N} = \{N_{p,q} : p, q \in Q\}$,
начальный нетерминал — $S = N_{q_1, q_0}$.

Множество грамматических правил состоит из
всех возможных правил вида

- ▶ $N_{q,q} \rightarrow \varepsilon$ для всех $q \in Q$;
- ▶ $N_{p,q} \rightarrow xN_{r,q}$ для всех переходов $(p, x, \varepsilon, r, \varepsilon)$,
где $p, q, r \in Q$, $x \in \Sigma \cup \{\varepsilon\}$;
- ▶ $N_{p,q} \rightarrow xN_{r,s}yN_{t,q}$ для всех пар переходов
 $(p, x, \varepsilon, r, Y)$ и $(s, y, Y, t, \varepsilon)$, где
 $p, q, r, s, t \in Q$, $x, y \in \Sigma \cup \{\varepsilon\}$, $Y \in \Gamma$.

МАГАЗИННЫЕ АВТОМАТЫ

Индукцией по длине вычисления (грамматического вывода) с применением Утверждения 8.4 можно показать, что для любых состояний $p, q \in Q$ и для любого слова w автомат \mathcal{B} имеет вычисление $(p, \varepsilon) \xrightarrow{w}^* (q, \varepsilon)$ в том и только том случае, если в грамматике $G_{\mathcal{B}}$ есть вывод $N_{p,q} \xrightarrow{G}^* w$.

Отсюда следует, что $L(\mathcal{B}) = L(G_{\mathcal{B}})$.

QED

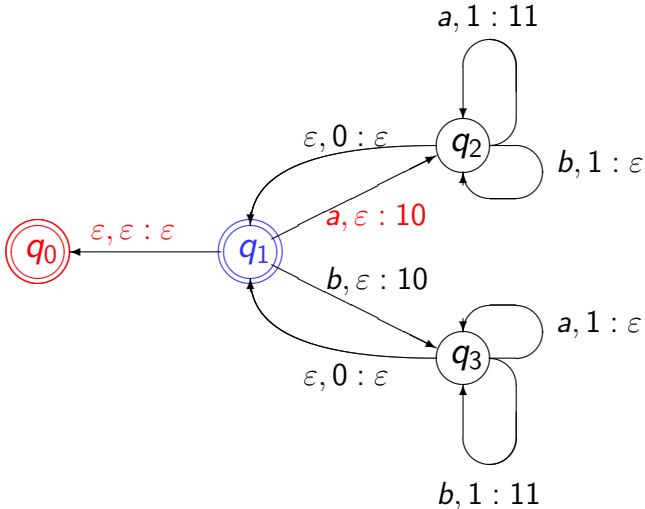
ДЕТЕРМИНИРОВАННЫЕ МАГАЗИННЫЕ АВТОМАТЫ

Недетерминированные автоматы не имеют прикладной ценности: их невозможно реализовать. Поэтому целесообразно рассмотреть детерминированный вариант магазинного автомата.

Магазинный автомат $\mathcal{B} = (\Sigma, \Gamma, Q, I, F, T)$ называется **детерминированным**, если для любой пары переходов $(q', x', Y', p', \alpha')$ и $(q'', x'', Y'', p'', \alpha'')$ выполняется хотя бы одно из трех требований:

1. $q' \neq q''$,
2. $x' \notin \text{Pref}(x'')$ и $x'' \notin \text{Pref}(x')$,
3. $y' \notin \text{Pref}(y'')$ и $y'' \notin \text{Pref}(y')$.

МАГАЗИННЫЕ АВТОМАТЫ



Недетерминированный автомат

ДЕТЕРМИНИРОВАННЫЕ МАГАЗИННЫЕ АВТОМАТЫ

Чтобы распознавать слова, детерминированному автомату нужно знать, где заканчивается слово. Поэтому будем предполагать, что каждое слово имеет на конце граничный маркер \dagger , $\dagger \notin \Sigma$.

Детерминированный автомат

$\mathcal{B} = (\Sigma \cup \{\dagger\}, \Gamma, Q, I, F, T)$ распознает язык L , $L \subseteq \Sigma^*$, если $L(\mathcal{B}) = \{w\dagger : w \in L\}$.

Пример. КС-язык $L = \{a^n b^n : n \geq 0\}$ является детерминированным КС-языком. Докажите, что язык $\{a, b\}^* \setminus L$ также является детерминированным КС-языком.

ДЕТЕРМИНИРОВАННЫЕ МАГАЗИННЫЕ АВТОМАТЫ

Перспектива автоматического построения синтаксических анализаторов КС-языков:

1. Описать синтаксис языка при помощи КС-грамматик (формулы Бэкуса-Наура).
2. Для заданной грамматики построить соответствующий магазинный автомат.
3. Детерминизируем этот магазинный автомат.
4. Минимизируем полученный детерминированный магазинный автомат.

Как решить задачи 3 и 4?

КОНЕЦ ЛЕКЦИИ 8