# Стековые автоматы

Теория формальных языков

*2021 г.*

## $\alpha$-преобразование

По-разному воспринимают переименовку:

- Переменные vs конструкторы в TRS;
- Нетерминалы vs терминалы в грамматиках.

## $\alpha$-преобразование

По-разному воспринимают переименовку:

- Переменные vs конструкторы в TRS;
- Нетерминалы vs терминалы в грамматиках.

Для любой инъективной $\sigma$ применение $\sigma$ к правилам грамматики/trs для переменных и нетерминалов также называется $\alpha$-преобразованием.

- $\alpha$-преобразование не меняет терминальный язык;
- обычно термы различаются с точностью до $\alpha$-преобразования.

## $\alpha$-преобразование

По-разному воспринимают переименовку:

- Переменные vs конструкторы в TRS;
- Нетерминалы vs терминалы в грамматиках.

Для любой инъективной $\sigma$ применение $\sigma$ к правилам грамматики/trs для переменных и нетерминалов также называется $\alpha$-преобразованием.

- $\alpha$-преобразование не меняет терминальный язык;
- обычно термы различаются с точностью до $\alpha$-преобразования.

Неформально: контейнеры определяются не именем, а содержимым (см. экстенсиональность в логике).

# Пересечение CFG и рег. языка

**Утверждение**

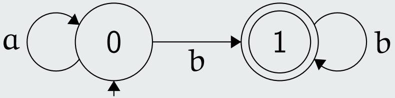
Даны CFG $\mathrm{G}$ и конечный автомат $\mathscr{A}$. Можно построить CFG $\mathrm{G}'$ такую, что $\mathrm{L}(\mathrm{G}') = \mathrm{L}(\mathrm{G}) \cap \mathrm{L}(\mathscr{A})$.

Предположим, что $\mathrm{G}$ — в k-нормальной форме Хомского, $\mathrm{q}$ — множество состояний автомата $\mathscr{A}$, $\mathrm{q_f}$ — единственное финальное состояние, $\mathrm{N}$ — множество нетерминалов грамматики $\mathrm{G}$. Множество нетерминалов $\mathrm{G}'$ — множество троек $\langle q_i, A, q_j \rangle$, $q_i, q_j \in q$, $A \in \mathrm{N}$.

- По каждому правилу $A \to A_1 \dots A_n$ из $\mathrm{G}$ строим правила $\langle p, A, q \rangle \to \langle p, A_1, q_1 \rangle \langle q_{n-1}, A_n, q \rangle$ для всех возможных $p$, $q$, $q_i$.
- По правилу вида $A \to t$ из $\mathrm{G}$ и переходу $p \to^{t} q$ строим правило $\langle p, A, q \rangle \to t$.
- Нетерминал $\langle q_0, S, q_f \rangle$ объявляем стартовым.

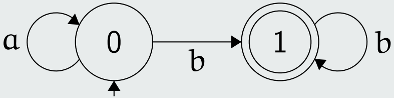
# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$,
$T \to b \mid SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:



# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$, $T \to b \mid SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:



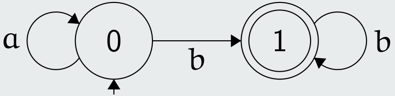

Сначала разберёмся с правилами вида $X \to t$. Если $t = a$, тогда подходящий нетерминал — только $G_A$, состояния — только 0+0. Если $t = b$, получается четыре комбинации состояний и нетерминалов.

$\langle 0, G_B, 1\rangle \to b \quad \langle 1, G_B, 1\rangle \to b$

$\langle 0, T, 1\rangle \to b \quad \langle 1, T, 1\rangle \to b \quad \langle 0, G_A, 0\rangle \to a$

# Пример

Построим пересечение языков CFG $S \to G_A T \,|\, SS$, $T \to b \,|\, SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:




$\langle 0, G_B, 1 \rangle \to b \quad \langle 1, G_B, 1 \rangle \to b$

$\langle 0, T, 1 \rangle \to b \quad \langle 1, T, 1 \rangle \to b \quad \langle 0, G_A, 0 \rangle \to a$

Рассмотрим возможные подстановки состояний в правила, порождаемые $S \to G_A T$, $S \to SG_B$. Соответствующие уравнения:

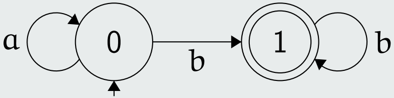
$\langle X1, S, X2 \rangle \to \langle X1, G_A, X3 \rangle \langle X3, T, X2 \rangle$

$\langle Y1, T, Y2 \rangle \to \langle Y1, S, Y3 \rangle \langle Y3, G_B, Y2 \rangle$

Чтобы правила были порождающими, необходимо положить $X1 = X3 = 0$, $Y2 = 1$. Выпишем все такие правила. Заметим, что получившийся в одном из них нетерминал $\langle 0, T, 0 \rangle$ — непорождающий, и удалим это правило.

# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$, $T \to b \mid SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:



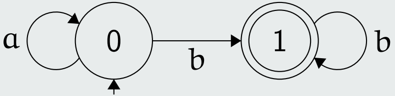

| | | |
|---|---|---|
| $\langle 0, G_B, 1\rangle \to b$ | $\langle 1, G_B, 1\rangle \to b$ | |
| $\langle 0, T, 1\rangle \to b$ | $\langle 1, T, 1\rangle \to b$ | $\langle 0, G_A, 0\rangle \to a$ |
| $\color{blue}{\langle 0, S, 0\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 0\rangle}$ | $\langle 0, S, 1\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 1\rangle$ | |
| $\langle 0, T, 1\rangle \to \langle 0, S, 0\rangle\langle 0, G_B, 1\rangle$ | $\langle 0, T, 1\rangle \to \langle 0, S, 1\rangle\langle 1, G_B, 1\rangle$ | |
| $\langle 1, T, 1\rangle \to \langle 1, S, 0\rangle\langle 0, G_B, 1\rangle$ | $\langle 1, T, 1\rangle \to \langle 1, S, 1\rangle\langle 1, G_B, 1\rangle$ | |

Осталось разобраться с правилами, порождёнными $S \to SS$.

Выпишем их общий вид: $\langle \color{red}{X1}, S, \color{red}{X2}\rangle \to \langle \color{red}{X1}, S, \color{red}{X3}\rangle\langle \color{red}{X3}, S, \color{red}{X2}\rangle$.

# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$, $T \to b \mid SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:




$\langle 0, G_B, 1\rangle \to b$ $\quad$ $\langle 1, G_B, 1\rangle \to b$

$\langle 0, T, 1\rangle \to b$ $\quad$ $\langle 1, T, 1\rangle \to b$ $\quad$ $\langle 0, G_A, 0\rangle \to a$

$\langle 0, S, 1\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 1\rangle$

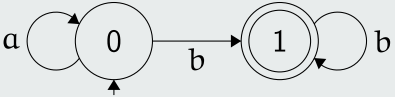
$\langle 0, T, 1\rangle \to \langle 0, S, 0\rangle\langle 0, G_B, 1\rangle$ $\quad$ $\langle 0, T, 1\rangle \to \langle 0, S, 1\rangle\langle 1, G_B, 1\rangle$

$\langle 1, T, 1\rangle \to \langle 1, S, 0\rangle\langle 0, G_B, 1\rangle$ $\quad$ $\langle 1, T, 1\rangle \to \langle 1, S, 1\rangle\langle 1, G_B, 1\rangle$

Осталось разобраться с правилами, порождёнными $S \to SS$.
Выпишем их общий вид: $\langle X1, S, X2\rangle \to \langle X1, S, X3\rangle\langle X3, S, X2\rangle$.
Если положить $X1 = 1$, $X2 = 0$, получим саморекурсивное правило $\langle 1, S, 0\rangle \to \alpha_1\langle 1, S, 0\rangle\alpha_2$. Но в построенной части грамматики нет правил вида $\langle 1, S, \dots\rangle \to \beta$. Поэтому нетерминал $\langle 1, S, 0\rangle$ — непорождающий. Удалим правила с его вхождением.

# Пример

Построим пересечение языков CFG $S \to G_A T \,|\, SS$, $T \to b \,|\, S G_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:




$\langle 0, G_B, 1\rangle \to b$ $\quad$ $\langle 1, G_B, 1\rangle \to b$

$\langle 0, T, 1\rangle \to b$ $\quad$ $\langle 1, T, 1\rangle \to b$ $\quad$ $\langle 0, G_A, 0\rangle \to a$

$\langle 0, S, 1\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 1\rangle$

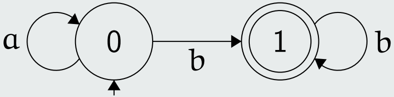
$\langle 0, T, 1\rangle \to \langle 0, S, 0\rangle\langle 0, G_B, 1\rangle$ $\quad$ $\langle 0, T, 1\rangle \to \langle 0, S, 1\rangle\langle 1, G_B, 1\rangle$

$\color{blue}{\langle 1, T, 1\rangle \to \langle 1, S, 1\rangle\langle 1, G_B, 1\rangle}$

Теперь если $\color{red}{X1} = \color{red}{X2} = 1$, то единственный вариант развёртки $S \to SS$ без участия нетерминала $\langle 1, S, 0\rangle$ будет иметь вид $\langle 1, S, 1\rangle \to \langle 1, S, 1\rangle\langle 1, S, 1\rangle$, так что нетерминал $\langle 1, S, 1\rangle$ тоже непорождающий.

# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$, $T \to b \mid S G_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:




$\langle 0, G_B, 1\rangle \to b$

$\langle 0, T, 1\rangle \to b$

$\langle 0, T, 1\rangle \to \langle 0, S, 0\rangle\langle 0, G_B, 1\rangle$

$\langle 1, G_B, 1\rangle \to b$

$\langle 1, T, 1\rangle \to b$ $\quad$ $\langle 0, G_A, 0\rangle \to a$

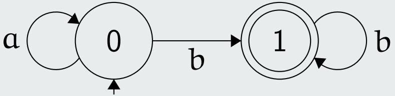
$\langle 0, S, 1\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 1\rangle$

$\langle 0, T, 1\rangle \to \langle 0, S, 1\rangle\langle 1, G_B, 1\rangle$

Аналогичным образом устанавливаем бесполезность нетерминала $\langle 0, S, 0\rangle$, который обязан ссылаться либо дважды на себя, либо на непорождающий $\langle 1, S, 0\rangle$.

# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$, $T \to b \mid SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:




$\langle 0, G_B, 1\rangle \to b$ $\quad\langle 1, G_B, 1\rangle \to b$

$\langle 0, T, 1\rangle \to b$ $\quad\langle 1, T, 1\rangle \to b$ $\quad\langle 0, G_A, 0\rangle \to a$

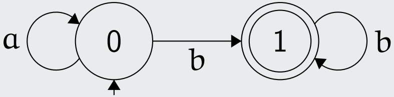
$$\langle 0, S, 1\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 1\rangle$$

$$\langle 0, T, 1\rangle \to \langle 0, S, 1\rangle\langle 1, G_B, 1\rangle$$

Теперь получается, что все варианты раскрытия нетерминала $\langle 0, S, 1\rangle$ по правилу $S \to SS$ включают непорождающие нетерминалы, поэтому никаких других правил в грамматику добавлять не надо. Осталось только удалить правила с недостижимыми нетерминалами $\langle 0, G_B, 1\rangle$, $\langle 1, T, 1\rangle$.

# Пример

Построим пересечение языков CFG $S \to G_A T \mid SS$, $T \to b \mid SG_B$, $G_A \to a$, $G_B \to b$, и следующего автомата:




$\langle 0, G_A, 0\rangle \to a$ $\quad$ $\langle 1, G_B, 1\rangle \to b$

$\langle 0, T, 1\rangle \to b$

$\langle 0, S, 1\rangle \to \langle 0, G_A, 0\rangle\langle 0, T, 1\rangle$ $\quad$ $\langle 0, T, 1\rangle \to \langle 0, S, 1\rangle\langle 1, G_B, 1\rangle$

Грамматика пересечения языков построена.

# Стековая память

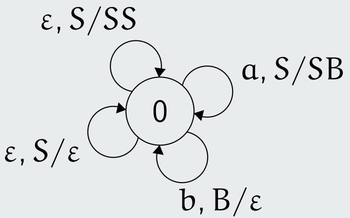
Пусть $G$ — CFG. Неформально представим, что $G$ — это стековый автомат, где состояния стека — нетерминальные сент. формы, порождаемые $G$. Скажем, что $G$ распознаёт только слова, соответствующие пустому стеку.

# Стековая память

Пусть $G$ — CFG. Неформально представим, что $G$ — это стековый автомат, где состояния стека — нетерминальные сент. формы, порождаемые $G$. Скажем, что $G$ распознаёт только слова, соответствующие пустому стеку.

## Грамматика и её стек

$$S \to aSB \,|\, SS \,|\, \varepsilon \qquad B \to b$$

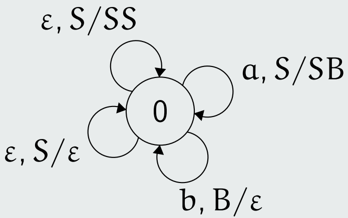


# Стековая память

Пусть $G$ — CFG. Неформально представим, что $G$ — это стековый автомат, где состояния стека — нетерминальные сент. формы, порождаемые $G$. Скажем, что $G$ распознаёт только слова, соответствующие пустому стеку.

## Грамматика и её стек

$$S \to aSB \mid SS \mid \varepsilon \qquad B \to b$$




А если в такие автоматы добавить ещё состояния?

# Pushdown Automata

## Определение

Стековый автомат $\mathscr{A}$ — кортеж $\langle \Pi, \Sigma, Q, \delta, q_0, Z_0 \rangle$, где:

- $\Pi$ — алфавит стека;
- $\Sigma$ — алфавит языка;
- $Q$ — множество состояний;
- $\delta$ — правила перехода вида $\langle q_i, t, P_i \rangle \to \langle q_j, \alpha \rangle$, где $t \in \Sigma \cup \{\varepsilon\}$, $\alpha \in \Pi^*$;
- $q_0$ — стартовое состояние, $Z_0$ – дно стека.

## Pushdown Automata

**Определение**

Стековый автомат $\mathscr{A}$ — кортеж $\langle \Pi, \Sigma, Q, \delta, q_0, Z_0 \rangle$, где:

- $\Pi$ — алфавит стека;
- $\Sigma$ — алфавит языка;
- $Q$ — множество состояний;
- $\delta$ — правила перехода вида $\langle q_i, t, P_i \rangle \to \langle q_j, \alpha \rangle$, где $t \in \Sigma \cup \{\varepsilon\}$, $\alpha \in \Pi^*$;
- $q_0$ — стартовое состояние, $Z_0$ – дно стека.

Два варианта допуска слова:

- если слово полностью прочитано, и стек пуст;
- если слово полностью прочитано, и состояние финальное.

# Виды допуска

**Утверждение**

PDA с допуском по конечному состоянию распознают те же языки, что и PDA с допуском по пустому стеку.

## Виды допуска

**Утверждение**

PDA с допуском по конечному состоянию распознают те же языки, что и PDA с допуском по пустому стеку.

- Пусть PDA допускает пустой стек. Добавим новый символ дна $Z_1$ и добавим по нему $\varepsilon$-переходы из всех состояний в новое финальное состояние.

## Виды допуска

**Утверждение**

PDA с допуском по конечному состоянию распознают те же языки, что и PDA с допуском по пустому стеку.

- Пусть PDA допускает пустой стек. Добавим новый символ дна $Z_1$ и добавим по нему $\varepsilon$-переходы из всех состояний в новое финальное состояние.
- Пусть PDA допускает финальные состояния. Добавим из них $\varepsilon$-переходы в состояние, опустошающее стек, а также новый символ стека $Z_1$ и новое стартовое состояние $q_0'$ с переходом $\langle q_0', \varepsilon, Z_0\rangle \to \langle q_0, Z_0 Z_1\rangle$.

# От CFG к PDA

**Утверждение**

По всякой CFG $\mathrm{G}$ можно построить PDA $\mathscr{A}$ такой, что $\mathrm{L}(\mathrm{G}) = \mathrm{L}(\mathscr{A})$.

# От CFG к PDA

**Утверждение**

По всякой CFG $G$ можно построить PDA $\mathscr{A}$ такой, что $L(G) = L(\mathscr{A})$.

Переведём $G$ в GNF и построим по ней PDA с единственным состоянием $0$ и допуском по пустому стеку, такой что $Z_0 = S$, правилу $A \to a$ соответствует переход $(0, a, A) \to (0, \varepsilon)$; правилу $A \to aB_1 \ldots B_n$ — переход $(0, a, A) \to (0, B_1 \ldots B_n)$.

## От PDA к CFG

**Утверждение**

По всякому PDA $\mathscr{A}$ можно построить CFG $\mathrm{G}$ такую, что $\mathrm{L}(\mathrm{G}) = \mathrm{L}(\mathscr{A})$.

# От PDA к CFG

**Утверждение**

По всякому PDA $\mathscr{A}$ можно построить CFG $G$ такую, что $L(G) = L(\mathscr{A})$.

Пусть $\mathscr{A}$ допускает слова по пустому стеку.

- Построим по стеку $\mathscr{A}$ вспомогательную $G'$:
  - введём новые стековые символы и заменим правила $(q_i, t, A) \to (q_j, A_1 \dots A_n)$ ($n \geqslant 1$) на пары $(q_i, \varepsilon, A) \to (q_i, A_0 \dots A_n)$, $(q_i, t, A_0) \to (q_j, \varepsilon)$.
  - переход $(q_i, \varepsilon, A) \to (q_j, A_0 A_1 \dots A_n)$ поставим в соответствие правилу $A \to A_0 A_1 \dots A_n$; переход $(q_i, t, A) \to (q_j, \varepsilon)$ поставим в соответствие правилу $A \to t_{i,j}$. $Z_0$ объявим стартовым символом. Пустой символ введём явно и так же пометим.

# От PDA к CFG

Пусть $\mathscr{A}$ допускает слова по пустому стеку.

- Построим по стеку $\mathscr{A}$ вспомогательную $G'$:
  - введём новые стековые символы и заменим правила $(q_i, t, A) \to (q_j, A_1 \dots A_n)$ $(n \geqslant 1)$ на пары $(q_i, \varepsilon, A) \to (q_i, A_0 \dots A_n)$, $(q_i, t, A_0) \to (q_j, \varepsilon)$.
  - переход $(q_i, \varepsilon, A) \to (q_j, A_0 A_1 \dots A_n)$ поставим в соответствие правилу $A \to A_0 A_1 \dots A_n$; переход $(q_i, t, A) \to (q_j, \varepsilon)$ поставим в соответствие правилу $A \to t_{i,j}$. $Z_0$ объявим стартовым символом. Пустой символ введём явно и так же пометим.
- Построим $\mathscr{A}'$ — FA с правилами вида $(q_i, t_{i,j}) \to q_j$, если для каких-нибудь $A$, $\alpha$ $(q_i, t, A) \to (q_j, \alpha)$ — переход $\mathscr{A}$. Все состояния объявим финальными.

# От PDA к CFG

Пусть $\mathscr{A}$ допускает слова по пустому стеку.

- Построим по стеку $\mathscr{A}$ вспомогательную $G'$:
  - введём новые стековые символы и заменим правила $(q_i, t, A) \to (q_j, A_1 \dots A_n)$ ($n \geqslant 1$) на пары $(q_i, \varepsilon, A) \to (q_i, A_0 \dots A_n)$, $(q_i, t, A_0) \to (q_j, \varepsilon)$.
  - переход $(q_i, \varepsilon, A) \to (q_j, A_0 A_1 \dots A_n)$ поставим в соответствие правилу $A \to A_0 A_1 \dots A_n$; переход $(q_i, t, A) \to (q_j, \varepsilon)$ поставим в соответствие правилу $A \to t_{i,j}$. $Z_0$ объявим стартовым символом. Пустой символ введём явно и так же пометим.
- Построим $\mathscr{A}'$ — FA с правилами вида $(q_i, t_{i,j}) \to q_j$, если для каких-нибудь $A$, $\alpha$ $(q_i, t, A) \to (q_j, \alpha)$ — переход $\mathscr{A}$. Все состояния объявим финальными.
- Теперь построим CFG — пересечение $G'$ и $\mathscr{A}'$ и сотрем все $\varepsilon_{i,j}$ и разметку терминалов. Грамматика $G$ готова!

# PDA в CFG формально

- Нетерминалы — тройки $[p, A, q]$, где $p, q \in Q$, $A \in \Pi$.
- По каждому переходу вида $(q, t, A) \to (p, A_1 \dots A_n)$ добавим правила для всех возможных $q_i$ вида $[q, A, q_n] \to t[p, A_1, q_1] \dots [q_{n-1}, A_n, q_n]$.
- По каждому переходу вида $(q, t, A) \to (p, \varepsilon)$ добавим правило $[q, A, p] \to t$.
- Разрешим стартовому состоянию переписываться в любое из $[q_0, Z_0, q]$.

# DPDA

**Определение**

PDA $\mathscr{A}$ детерминированный, если:

- если есть переход $\langle q, \varepsilon, Z \rangle \to \ldots$, то больше никаких переходов по $Z$ из состояния $q$ нет;
- каждой тройке $\langle q, a, Z \rangle$, $a \in \Sigma$, соответствует не больше одной правой части.

# DPDA

**Определение**

PDA $\mathscr{A}$ детерминированный, если:

- если есть переход $\langle q, \varepsilon, Z\rangle \to \ldots$, то больше никаких переходов по $Z$ из состояния $q$ нет;
- каждой тройке $\langle q, a, Z\rangle$, $a \in \Sigma$, соответствует не больше одной правой части.

DPDA слабее, чем NPDA. Например, язык $\{a^n b^m] \mid n = m \vee m = 2 * n\}$ не распознается DPDA.

# DPDA

**Определение**

PDA $\mathscr{A}$ детерминированный, если:

- если есть переход $\langle q, \varepsilon, Z\rangle \to \dots$, то больше никаких переходов по $Z$ из состояния $q$ нет;
- каждой тройке $\langle q, a, Z\rangle$, $a \in \Sigma$, соответствует не больше одной правой части.

DPDA слабее, чем NPDA. Например, язык $\{a^n b^m] \,|\, n = m \vee m = 2 * n\}$ не распознается DPDA. DPDA с допуском по пустому стеку ещё слабее — язык $\{a^n\}$ не может быть распознан DPDA с таким допуском.

# Двухсторонние PDA

**Утверждение**

Двухсторонние PDA распознают больше языков, чем односторонние.

Доказательство: язык $\{a^n b^n c^n\}$ распознаваем двухсторонним PDA.