

Стековые автоматы



Теория формальных языков
2021 г.



Свойства замкнутости КС-языков

Пусть стартовый нетерминал грамматики G_i — это S_i .

- КС-языки тривиально замкнуты относительно объединения и конкатенации. Объединение: добавим правило $S' \rightarrow S_1 \mid S_2$, конкатенация: добавим правило $S' \rightarrow S_1 S_2$.
- КС-языки замкнуты относительно реверсирования (достаточно реверсировать левые части правил), а также префиксных и суффиксных замыканий (упражнение после освоения материала по PDA, когда-нибудь в будущем и здесь).



Свойства замкнутости КС-языков

- КС-языки не замкнуты относительно пересечения.
Универсальный контрпример: пусть $\$$ – символ-разделитель (отсутствует в словаре). Рассмотрим язык $\{w_1\$(.)^*\$w_3\}$, где слова w_1 и w_2 связаны порождающими правилами (т.е. структура $w_1\$w_3$ не регулярна), и язык $\{w_1\$w_2\$(.)^*\}$, где такими правилами связаны w_1 и w_2 . Их пересечение вынудит существование нерегулярной зависимости между w_1 и одновременно w_2 и w_3 , что порождает не две, как в КС-языках, а минимум три области накачки.

Конкретные примеры: пара $L_1 = \{a^n\$a^n\$a^*\}$,
 $L_2 = \{a^n\$a^*\$a^n\}$; или пара $L'_1 = \{w\$w^R\$a^*\}$,
 $L'_2 = \{w\$(a|b)^*\$a^n \mid |w|_a = n\}$.



Свойства замкнутости КС-языков

- КС-языки не замкнуты относительно дополнения. В противном случае пересечение также не нарушало бы свойство контекстной свободы. Контпримеры строятся на основе этой же идеи: берём язык-пересечение КС-языков L_1 и L_2 , не являющийся КС. Чаще всего его дополнение — КС-язык.

Конкретные примеры: КС-язык

$L' = \overline{L_1 \cap L_2} = \{a^m \$ a^n \$ a^k \mid m \neq n \vee m \neq k \vee n \neq k\}$ с не

КС-дополнением; КС-язык

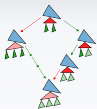
$L'' = \overline{L'_1 \cap L'_2} = \{w \$ v \$ a^n \mid v \neq w^R \vee n \neq |w|_a\}$ с не

КС-дополнением.



Свойства замкнутости КС-языков

- КС-языки тривиально замкнуты относительно объединения и конкатенации.
- КС-языки замкнуты относительно реверсирования (достаточно реверсировать левые части правил), а также префиксных и суффиксных замыканий (упражнение после освоения материала по PDA, когда-нибудь в будущем и здесь).
- КС-языки не замкнуты относительно пересечения.
- КС-языки не замкнуты относительно дополнения.
- КС-свободные языки замкнуты относительно пересечения с регулярным языком (см. ниже).



Пересечение КС-грамматики и рег. языка

Утверждение

Даны КС-грамматика G и конечный автомат \mathcal{A} . Можно построить КС-грамматику G' такую, что $L(G') = L(G) \cap L(\mathcal{A})$.

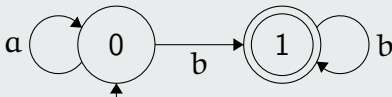
Предположим, что G — в k -нормальной форме Хомского, q — множество состояний автомата \mathcal{A} , q_f — единственное финальное состояние, N — множество нетерминалов грамматики G . Множество нетерминалов G' — множество троек $\langle q_i, A, q_j \rangle$, $q_i, q_j \in q$, $A \in N$.

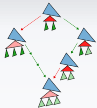
- По каждому правилу $A \rightarrow A_1 \dots A_n$ из G строим правила $\langle p, A, q \rangle \rightarrow \langle p, A_1, q_1 \rangle \langle q_{n-1}, A_n, q \rangle$ для всех возможных p, q, q_i .
- По правилу вида $A \rightarrow t$ из G и переходу $p \xrightarrow{t} q$ строим правило $\langle p,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



Пример

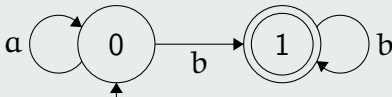
Построим пересечение языков CFG $S \rightarrow G_A T \mid SS$,
 $T \rightarrow b \mid S G_B$, $G_A \rightarrow a$, $G_B \rightarrow b$, и следующего автомата:





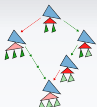
Пример

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



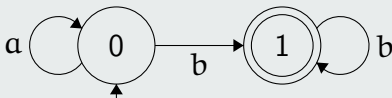
Сначала разберёмся с правилами вида $X \rightarrow t$. Если $t = a$, тогда подходящий нетерминал — только G_A , состояния — только $0+0$. Если $t = b$, получается четыре комбинации состояний и нетерминалов.

$$\begin{array}{lll} \langle 0, G_B, 1 \rangle \rightarrow b & \langle 1, G_B, 1 \rangle \rightarrow b & \\ \langle 0, T, 1 \rangle \rightarrow b & \langle 1, T, 1 \rangle \rightarrow b & \langle 0, G_A, 0 \rangle \rightarrow a \end{array}$$



Пример

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



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

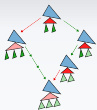
$$\langle 0, T, 1 \rangle \rightarrow b \quad \langle 1, T, 1 \rangle \rightarrow b \quad \langle 0, G_A, 0 \rangle \rightarrow a$$

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

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

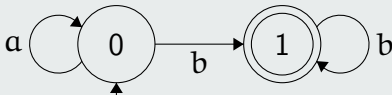
$$\langle Y1, T, Y2 \rangle \rightarrow \langle Y1, S, Y3 \rangle \langle Y3, G_B, Y2 \rangle$$

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 $S \rightarrow G_A T \mid SS$,
 $T \rightarrow b \mid S G_B$, $G_A \rightarrow a$, $G_B \rightarrow b$, и следующего автомата:



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

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

$\langle 0, S, 0 \rangle \rightarrow \langle 0, G_A, 0 \rangle \langle 0, T, 0 \rangle$

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

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

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

$\langle 1, T, 1 \rangle \rightarrow b$

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

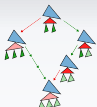
$\langle 0, T, 1 \rangle \rightarrow \langle 0, S, 1 \rangle \langle 1, G_B, 1 \rangle$

$\langle 1, T, 1 \rangle \rightarrow \langle 1, S, 1 \rangle \langle 1, G_B, 1 \rangle$

$\langle 0, G_A, 0 \rangle \rightarrow a$

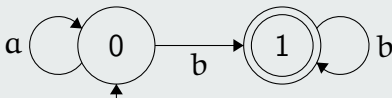
Осталось разобраться с правилами, порождёнными $S \rightarrow SS$.

Выпишем их общий вид: $\langle X1, S, X2 \rangle \rightarrow \langle X1, S, X3 \rangle \langle X3, S, X2 \rangle$.



Пример

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



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

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

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

$$\langle 1, T, 1 \rangle \rightarrow b$$

$$\langle 0, G_A, 0 \rangle \rightarrow a$$

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

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

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

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

$$\langle 1, T, 1 \rangle \rightarrow \langle 1, S, 1 \rangle \langle 1, G_B, 1 \rangle$$

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

Выпишем их общий вид: $\langle X1, S, X2 \rangle \rightarrow \langle X1, S, X3 \rangle \langle X3, S, X2 \rangle$.

Если положить $X1 = 1$, $X2 = 0$, получим саморекурсивное правило

$\langle 1, S, 0 \rangle \rightarrow \alpha_1 \langle 1, S, 0 \rangle \alpha_2$. Но в построенной части грамматики нет

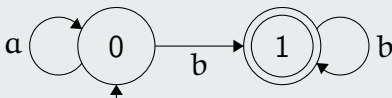
правил вида $\langle 1, S, \dots \rangle \rightarrow \beta$. Поэтому нетерминал $\langle 1, S, 0 \rangle$ —

непорождающий. Удалим правила с его вхождением.



Пример

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



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

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

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

$$\langle 1, T, 1 \rangle \rightarrow b$$

$$\langle 0, G_A, 0 \rangle \rightarrow a$$

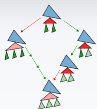
$$\langle 0, S, 1 \rangle \rightarrow \langle 0, G_A, 0 \rangle \langle 0, T, 1 \rangle$$

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

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

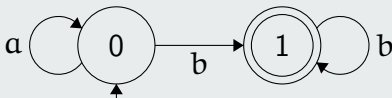
$$\langle 1, T, 1 \rangle \rightarrow \langle 1, S, 1 \rangle \langle 1, G_B, 1 \rangle$$

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



Пример

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



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

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

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

$$\langle 1, T, 1 \rangle \rightarrow b$$

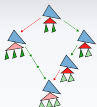
$$\langle 0, G_A, 0 \rangle \rightarrow a$$

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

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

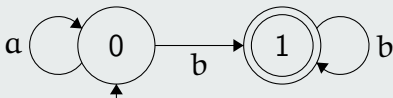
$$\langle 0, T, 1 \rangle \rightarrow \langle 0, S, 1 \rangle \langle 1, G_B, 1 \rangle$$

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 $S \rightarrow G_A T \mid SS$,
 $T \rightarrow b \mid S G_B$, $G_A \rightarrow a$, $G_B \rightarrow b$, и следующего автомата:



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

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

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

$\langle 1, T, 1 \rangle \rightarrow b$

$\langle 0, G_A, 0 \rangle \rightarrow a$

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

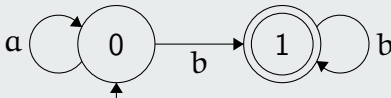
$\langle 0, T, 1 \rangle \rightarrow \langle 0, S, 1 \rangle \langle 1, G_B, 1 \rangle$

Теперь получается, что все варианты раскрытия нетерминала $\langle 0, S, 1 \rangle$ по правилу $S \rightarrow SS$ 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 $S \rightarrow G_A T \mid SS$,
 $T \rightarrow b \mid S G_B$, $G_A \rightarrow a$, $G_B \rightarrow b$, и следующего автомата:



$\langle 0, G_A, 0 \rangle \rightarrow a$

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

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

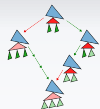
$\langle 0, S, 1 \rangle \rightarrow \langle 0, G_A, 0 \rangle \langle 0, T, 1 \rangle$ $\langle 0, T, 1 \rangle \rightarrow \langle 0, S, 1 \rangle \langle 1, G_B, 1 \rangle$

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.



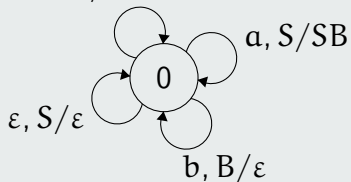
Стековая память

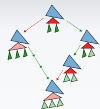
Пусть G — CFG. Неформально представим, что G — это стековый автомат, где состояния стека — нетерминальные сент. формы, порождаемые G . Скажем, что G распознаёт только слова, соответствующие пустому стеку.

Грамматика и её стек

$$S \rightarrow \alpha SB \mid SS \mid \varepsilon \quad B \rightarrow b$$

$\varepsilon, S/SS$





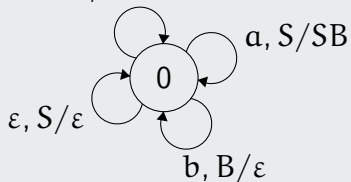
Стековая память

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

$$S \rightarrow \alpha SB \mid SS \mid \varepsilon \quad B \rightarrow b$$

$\varepsilon, S/SS$



А если в такие автоматы добавить ещё состояния?



Pushdown Automata

Определение

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

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



Pushdown Automata

Определение

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

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

Два варианта допуска слова:

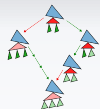
- если слово полностью прочитано, и стек пуст;
- если слово полностью прочитано, и состояние финальное.



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 ε -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.

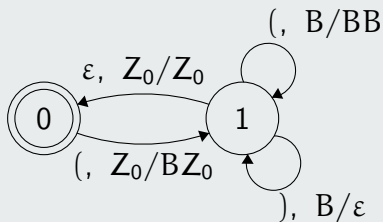
- Пусть PDA допускает пустой стек. Добавим новый символ дна Z_1 и добавим по нему ε -переходы из всех состояний в новое финальное состояние.
- Пусть PDA допускает финальные состояния. Добавим из них ε -переходы в состояние, опустошающее стек, а также новый символ стека Z_1 и новое стартовое состояние q'_0 с переходом $\langle q'_0, \varepsilon, Z_0 \rangle \rightarrow \langle q_0, Z_0 Z_1 \rangle$.



Пример оформления PDA

Обычно PDA изображается в виде автомата, в котором стрелки помечены сигнатурой $\alpha, T/\Phi$, где α — это символ терминального алфавита (или пустое слово), T — символ на вершине стека, Φ — последовательность стековых символов, помещаемая на вершину стека вместо T .

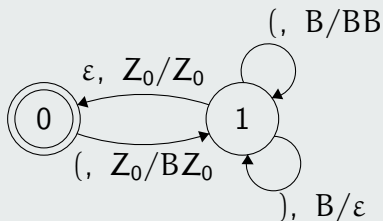
Следующий PDA распознаёт правильные скобочные последовательности (включая пустое слово).





Пример оформления PDA

Следующий PDA распознаёт правильные скобочные последовательности (включая пустое слово).



Заметим, что перехода из состояния 0 по символу $)$ нет. Так же как и в случае конечных автоматов, можно добавить для такого перехода состояние-ловушку, потому что он порождает слово, в префиксе которого количество закрывающих скобок превышает количество открывающих, а такие слова не являются ПСП.



От CFG к PDA

Утверждение

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



От CFG к PDA

Утверждение

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

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



От PDA к CFG

Утверждение

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



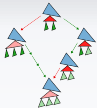
От PDA к CFG

Утверждение

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

Пусть \mathcal{A} допускает слова по пустому стеку.

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



От PDA к CFG

Пусть \mathcal{A} допускает слова по пустому стеку.

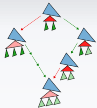
- Построим по стеку \mathcal{A} вспомогательную G' :
 - введём новые стековые символы и заменим правила $(q_i, t, A) \rightarrow (q_j, A_1 \dots A_n)$ ($n \geq 1$) на пары $(q_i, \varepsilon, A) \rightarrow (q_i, A_0 \dots A_n)$, $(q_i, t, A_0) \rightarrow (q_j, \varepsilon)$.
 - переход $(q_i, \varepsilon,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



От PDA к CFG

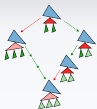
Пусть \mathcal{A} допускает слова по пустому стеку.

- Построим по стеку \mathcal{A} вспомогательную G' :
 - введём новые стековые символы и заменим правила $(q_i, t, A) \rightarrow (q_j, A_1 \dots A_n)$ ($n \geq 1$) на пары $(q_i, \varepsilon, A) \rightarrow (q_i, A_0 \dots A_n)$, $(q_i, t, A_0) \rightarrow (q_j, \varepsilon)$.
 - переход $(q_i, \varepsilon, A) \rightarrow (q_j, A_0 A_1 \dots A_n)$ поставим в соответствие правилу $A \rightarrow A_0 A_1 \dots A_n$; переход $(q_i, t, A) \rightarrow (q_j, \varepsilon)$ поставим в соответствие правилу $A \rightarrow t_{i,j}$. Z_0 объявим стартовым символом. Пустой символ введём явно и так же пометим.
- Построим \mathcal{A}' — FA с правилами вида $(q_i, t_{i,j}) \rightarrow q_j$, если для каких-нибудь A, α $(q_i, t, A) \rightarrow (q_j, \alpha)$ — переход \mathcal{A} . Все состояния объявим финальными.
- Теперь построим CFG — пересечение G' и \mathcal{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 $(q, t, A) \rightarrow (p, A_1 \dots A_n)$ добавим правила для всех возможных q_i вида $[q, A, q_n] \rightarrow t[p, A_1, q_1] \dots [q_{n-1}, A_n, q_n]$.
- По каждому переходу вида $(q, t, A) \rightarrow (p, \varepsilon)$ добавим правило $[q, A, p] \rightarrow t$.
- Разрешим стартовому состоянию переписываться в любое из $[q_0, Z_0, q]$.



DPDA

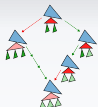
Определение

PDA \mathcal{A} детерминированный, если:

- если есть переход $\langle q, \varepsilon, Z \rangle \rightarrow \dots$, 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 $\{a^n b^m \mid n = m \vee m = 2 * n\}$ не распознается DPDA.

DPDA с допуском по пустому стеку ещё слабее — язык $\{a^n\}$ не может быть распознан DPDA с таким допуском.



DPDA

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

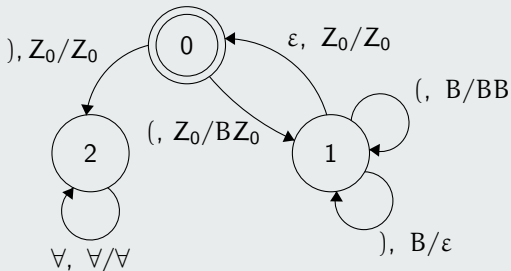
Предположим, что существует DPDA, распознающий язык $\{a^n b^m \mid n = m \vee m = 2 * n\}$. Тогда после чтения префикса $a^n b^n$ слова $a^n b^{2n}$ он должен находиться в финальном состоянии. Далее он должен распознать ровно n букв b . Заменяем часть автомата, распознающую этот фрагмент слова, на изоморфную ей, но читающую только буквы c . Получим PDA, распознающий язык $\{a^n b^n\} \cup \{a^n b^n c^n\}$, не являющийся КС.

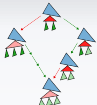
Предположим, что существует DPDA с допуском по пустому стеку, распознающий язык $\{a^n\}$. Тогда на слове a стек этого автомата должен быть уже точно пуст \Rightarrow в этом состоянии вообще невозможно сделать дальнейшие переходы.



Пример DPDA

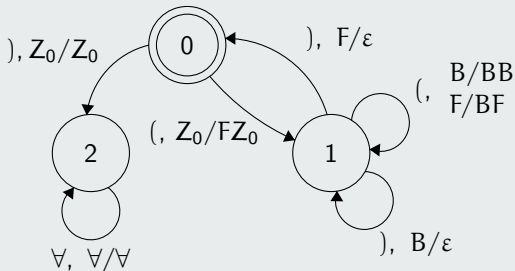
PDA для ПСП, приведённый выше, является DPDA, в чём нетрудно убедиться, проверив, что ε -переход совершается лишь в том случае, когда никакие другие совершить невозможно. Добавим в него состояние-ловушку.

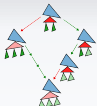




Пример DPDA

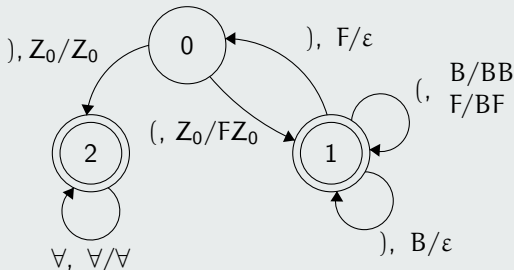
Чтобы не описывать многочисленные переходы из состояния-ловушки в себя по всем парам «символ ленты — символ стека», мы воспользовались сокращённым обозначением $\forall, \forall/\forall$, подразумевающая следующее: «по любой паре \langle терминал, символ стека \rangle в состоянии 2 переходим в себя, сохраняя символ стека на вершине». Также избавимся от ε -перехода, введя символ стека F , т.е. «самая первая скобка».





Пример DPDA

Поскольку автомат \mathcal{A} — детерминированный, в нём существуют переходы по всем комбинациям $\langle \text{терминал, символ стека} \rangle$, и нет ε -переходов, связывающих нефинальное и финальное состояния, то автомат, в котором все конечные состояния \mathcal{A} заменены на нефинальные и наоборот, распознаёт дополнение языка, распознаваемого PDA \mathcal{A} . Значит, мы показали, что дополнение языка ПСП контекстно-свободно, и предъявили PDA, который распознаёт его.





Двухсторонние PDA

Утверждение

Двухсторонние PDA распознают больше языков, чем односторонние.

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