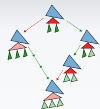


Стековые автоматы



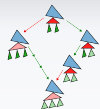
Теория формальных языков
2021 г.



α -преобразование

По-разному воспринимают переименовку:

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



α -преобразование

По-разному воспринимают переименовку:

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

Для любой инъективной σ применение σ к правилам грамматики/trs для переменных и нетерминалов также называется α -преобразованием.

- α -преобразование не меняет терминальный язык;
- обычно термы различаются с точностью до α -преобразования.



α -преобразование

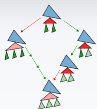
По-разному воспринимают переименовку:

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

Для любой инъективной σ применение σ к правилам грамматики/trs для переменных и нетерминалов также называется α -преобразованием.

- α -преобразование не меняет терминальный язык;
- обычно термы различаются с точностью до α -преобразования.

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



Пересечение CFG и FA

Утверждение

Даны CFG G и FA \mathcal{A} . Можно построить CFG G' такую, что $L(G') = L(G) \cup L(\mathcal{A})$.

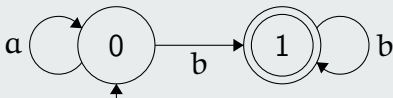
Предположим, что G — в k-CNF, q — множество состояний \mathcal{A} , q_f — единственное финальное состояние, N — множество нетерминалов G . Множество нетерминалов G' — множество троек $\langle q_i, A, q_j \rangle$, $q_i, q_j \in q$, $A \in N$.

- 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 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:





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.



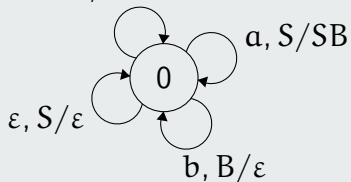
Стековая память

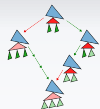
Пусть G — CFG. Неформально представим, что G — это стековый автомат, где состояния стека — нетерминальные сент. формы, порождаемые G . Скажем, что G распознаёт только слова, соответствующие пустому стеку.

Грамматика и её стек

$$S \rightarrow aSB \mid SS \mid \varepsilon \quad B \rightarrow b$$

$\varepsilon, S/SS$





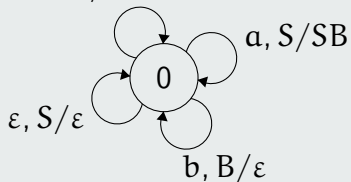
Стековая память

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.

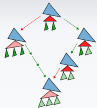
Грамматика и её стек

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

$\varepsilon, S/SS$



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 $\langle q'_0, \varepsilon, Z_0 \rangle \rightarrow \langle q_0, Z_0 Z_1 \rangle$.



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 $[q_0, Z_0, q]$.

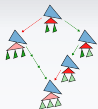


DPDA

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.



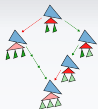
DPDA

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

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.



Двухсторонние PDA

Утверждение

Двухсторонние PDA распознают больше языков, чем односторонние.

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