Язык стека автомата с магазинной памятью – регулярен

$I = \{\alpha | \exists \omega : (q_0, \omega, \varepsilon) \vdash^* (q_1, \beta, \alpha) \vdash^* (q_2, \varepsilon, \varepsilon); q_1, q_2 \in Q_M\}$ *- регулярен*

Автор: Плюснин Павел, студент 2 курса МФТИ
Аннотация: в данной статье будет приведено доказательство того, что язык МП автомата является регулярным. В статье будет введена новая структура - префиксный автомат, будет рассмотрен алгоритм получения префиксной грамматики из префиксного автомата. Новая структура, возможно, является более удобным способом задания регулярных языков

Пусть нам дан МП-автомат M. В нем все переходы имеют вид $\sigma, \tau/\alpha$ (строка такого вида стоит над стрелкой). Построим новый "автомат"P, отличающийся от M лишь функцией перехода. Теперь над каждой стрелцой вместо $\sigma, \tau/\alpha$ напишем $\tau \to \alpha$. Автомат такого вида будем называть префиксным.

Определение. *Префиксный автомат* - $(\Gamma, Q, q_0, Z, P, \delta)$*, где*

- Γ - *алфавит вывода*
- Q - *множество состояний автомата*
- $q_0 \in Q$ - *начальное состояние автомата*
- Z - *начальный символ автомата*
- P - *множество правил вывода в виде* $u \to v$*, где* $u, v \in \Gamma^*$
- $\delta : Q \times P \to 2^Q$ - *функция переходов*

Префиксный автомат не имеет входа, он не проверяет принадлежность слова языку, он сам генерирует из начального символа множество слов языка. Язык, выводимый данным автоматом - это множество всех строк, выводимое из Z за ноль и более шагов, применяя правила соответствующих состояний, при этом совершая переходы согласно функции переходов.

Префиксный автомат P назовем префиксным автоматом для МП-автомата M, если язык автомата P совпадает с множеством всех слов, получающихся в стеке автомата M на некотором этапе вывода (возможно, неудачного) некоторого слова.
Очевидно, что построенный выше автомат P - префиксный автомат для M, $I \subseteq L(P)$

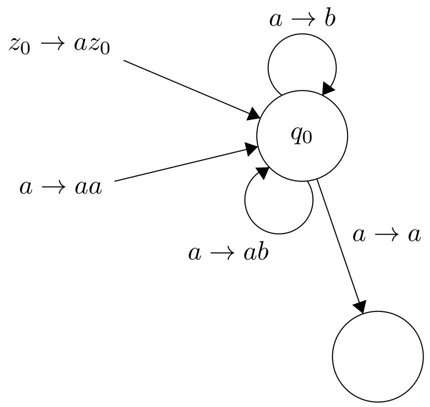
Автомат U получим из автомата P удалением всех правил, которые впоследстиве не смогут привести к опустошению стека (обращению выводимого слова в ε).
Все правила и переходы автомата S, язык которого совпадает с I, уже содержатся в автомате P, возможно, надо лишь убрать из P некоторые правила.

Но, возможно, в префиксном автомате остались некоторые последовательности правил, применяя которые, слово не будет принадлежать I (его уже нельзя будет обратить в ε). Очевидно, что каждую последовательность χ можно раздуть в бесконеченое число других последовательностей, в целый класс, получаемых из χ добавлением других правил. Будем рассматривать только последовательности минимальной длины в своем

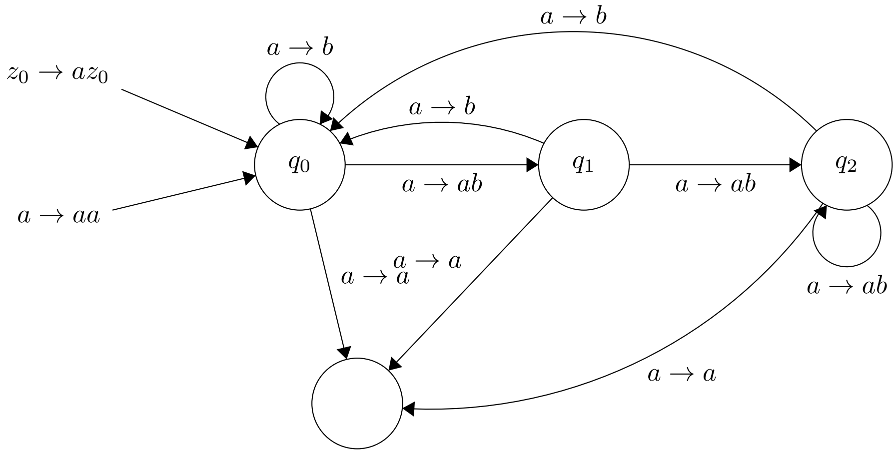
классе. Таких последовательностей конечное число.
Пусть $\chi = p_1 p_2 \dots p_n$ - такая последовательность длины n. Т.е если применить правила $p_1 \dots p_n$ именно в таком порядке (возможно, применяя между ними еще какие-то правила), выводимая цепочка уже не будет принадлежать I. Построим вспомогательный префиксный автомат S' для χ:

- Для каждого состояния q_0:
  Для каждой петли, повторяющейся в последовательности χ $k > 0$ раз:
  - добавим к состоянию q еще k состояний
  - уберем из q_0 рассматриваемую петлю, последовательно свяжем добавленые состояния правилом петли
  - все правила, выходящие из q_0 продублируем и в новых состояниях
  - добавим рассматриваемую петлю в q_k - последнее добавленное состояние

Например q_0 при рассматриваемой петле $a \to ab$



Для $K = 2$ преобразуется в



Преобразованный автомат эквивалентен исходному - просто в нем некоторые петли расписаны несколько раз в явном виде.

Тогда последовательно для каждого χ рассмотрим все возможные префиксные автоматы, в которых отсутствует некоторое количество звеньев последовательности. Объединение языков полученных префиксных автоматов и задает язык I.

Докажем, что префиксные автоматы задают регулярные языки:

Определение. *Функция $first(\alpha)$ возвращает первую букву строки α*

Будем обозначать состояния натуральными числами. Также будем считать, что начальный символ Z приходит в начальное состояние 1 из фиктивного состояния 0.

Определение. *Пусть $IN(n)$ – множество всех $first(\beta)$ - множество всех возожных первых букв β правил $\alpha \to \beta$, ведущих в n, с индексом, равным состоянию k, из которого данное правило идет*

Определение. *Грамматически корректным префиксным автоматом назовем такой префиксный автомат, полученный после применения следующего алгоритма*

- *Для всех состояний $n \in Q$ (здесь работаем только с правилами, выходящими из n):*
  1. *Для каждого $x_k \in IN(n)$ ко всем правилам вида $x \to \alpha$ добавляем правило $x_k \to \alpha_n$.*
  2. *Для каждого правила вида $x \to \alpha$ (т.е слева у буквы индекса нет) добавляем правило $x_{(n)} \to \alpha_n$, если $\alpha \neq \varepsilon$ удаляем правило $x \to \alpha$*
  3. *Для всех правил вида $w \to \varepsilon$ (где w может как иметь индекс, так и нет):*
     - *$\forall \sigma \in \Gamma, \forall k \in Q$ добаялем правило $w\sigma_k \to \sigma_{(n)}$, если это правило принадлежит петле (ведет в то же состояние, откуда и выходит), и $w\sigma_k \to \sigma_n$ в ином случае*
     - *Добавляем правило $\alpha \to e$, где e - символ, которого раньше не было в алфавите автомата до алгоритма ($e \notin \Gamma$)*
     - *Удаляем правило $w \to \varepsilon$*

Построим из префиксного автомата S ГКПА T.

Заметим, что полученный автомат T выводит почти тот же язык, что и исходный, только в нем много синонимов - буквы из Γ_S имеют множество своих аналогов с индексами в Γ_T.

По ГКПА T построим префиксную грамматику.

Определение. *Префиксная грамматика - это тройка (Γ, Z, P), где*

- Γ *- конечный алфавит*
- Z *- множество основных строк над* Γ
- P *- множество правил вывода в виде $u \to v$, где $u, v \in \Gamma^*$*

Префиксную грамматику G' получаем из T следующим образом:

- $\Gamma = \{x_i | x \in \Gamma_T, i \in Q_T\}$
- $Z = \{Z_T\}$

- $P = P_T$

Т.к множество Z состоит лишь из одного элемента, так что под Z теперь будем понимать элемент Z_T.

Мы получили префиксную грамматику G', а значит $L(G')$ - регулярный, а значит для него существует некоторое регулярное выражение.
Заметим, что полученный язык почти совпадает с искомым, просто многие буквы из I имеют множество своих аналогов с индексами в G'. Не будем различать одинаковые буквы, но с различными индексами (или вовсе без них), т.е $x_i = x$. Получим необходимый язык I, который является регулярным, так как для него тоже существует регулярное выражение (просто заменим в регулярном вырвжении для $L(G')$ все x_i на x). $\square$

Приложение. Префиксные грамматики описывают регулярные языки.[1]

Алгоритм преобразования префиксной грамматики к праволинейной

Вход: префиксная грамматика G_p с основными строками $\alpha_1 \dots \alpha_n$ и правилами $\beta_1 \to \gamma_1, \dots, \beta_m \to \gamma_m$
Выход: праволинейная грамматика G_R
Пусть G_R - праволинейная грамматика с аксиомой S, нетерминалами $V_{\beta_1} \dots V_{\beta_m}$, и всеми нетерминалами из G_p. G_R содержит следующие правила:
$S \to \alpha_1 | \dots | \alpha_n | \gamma_1 V_{\beta_1} | \dots | \gamma_m V_{\beta_m}$.
Также, будем пока возможно добавлять в грамматику следующие правила:
$V_{\beta_i} \to \omega V_{\beta_j}$, где
$S \to \psi_1 V_{\beta_{i_1}}, V_{\beta_{i_1}} \to \psi_2 V_{\beta_{i_2}}, \dots, V_{\beta_{i_{k-1}}} \to \psi_k \omega V_{\beta_j}$ - правила уже в грамматике, $\psi_1, \dots, \psi_k, \omega$ - строки, причем $\psi_1 \dots \psi_k = \beta_i$, V_{β_j} - нетерминал или ε .

Список использованной литературы:

1. Prefix grammars: An alternative characterization of the regular languages. Michael Frazier, C. David Page Jr. Information Processing Letters

[1] Prefix grammars: An alternative characterization of the regular languages. Michael Frazier, C. David Page Jr. Information Processing Letters