Задание 7

Контекстно-свободные языки и магазинные автоматы

Ключевые слова [1]:*язык, контекстно-свободный язык, магазинный автомат, грамматика, метод математической индукции.*

1 МП-автоматы

1.1 Определения

Моделью вычислений, распознающей класс контекстно-свободных языков (CFL), является автомат с магазинной памятью.

Под магазинной памятью понимается стек. Вы должны быть знакомы со стеком из курса информатики, но на всякий случай я скажу про него пару слов. Неформально, стек – это стопка, например такая как колода карт. Работать со стеком можно следующим образом: карты можно брать только последовательно с верхушки колоды – если вы хотите вытащить вторую карту, то сначала вы должны взять первую, класть карты можно только на верх колоды. Сколько карт в колоде вы не знаете, а знаете только пуста колода или нет.

Формально, под стеком понимается следующая структура данных:

- элементы стека располагаются в порядке добавления, первый элемент, добавленный в стек называется *дном*, последний элемент, добавленный в стек, называется *верхушкой*;

- операция $push(a)$ добавляет элемент a в стек, причём a становится верхушкой стека;

- операция $pop()$ возвращает элемент a, находящийся на верхушке стека;

- операция $empty()$ проверяет пустоту стека и возвращает истинное значение, если стек пуст.

[1]минимальный необходимый объем понятий и навыков по этому разделу)

Магазинные автоматы встречаются куда более чаще, чем стековые, но называются они магазинными, потому что на заре этой науки кто-то застолбил название стековые автоматы и под ними стали пониматься такие автоматы со стеком, что автомат мог просматривать содержимое стека, не меняя его содержания, а менять его только согласно правилам работы со стеком, что сильно меняло класс языков, распознаваемых автоматом: например, язык $a^n b^n c^n$ не распознаётся ни одним МП-автоматом, но распознаётся стековым автоматом.

Теперь дадим формальное определение автомату с магазинной памятью.

Определение 1. Магазинный автомат содержит семь компонент и выглядит следующим образом: $P = (\Sigma, \Gamma, Q, q_0, Z_0, \delta, F)$, где

- Σ – входной алфавит;
- Γ – алфавит стека, т.е. символы, которые можно добавлять в стек;
- Q – множество состояний автомата;
- $q_0 \in Q$ – начальное состояние автомата;
- $Z_0 \in \Gamma$ – единственный символ, находящийся в стеке при начале работы автомата;
- $\delta : Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \to 2^{Q \times \Gamma^*}$ – функкия переходов;
- F – множество принимающих состояний.

В начале работы автомат находится в состоянии q_0 и в магазине лежит только символ Z_0. за такт работы, автомат считывает букву из входного слова (или же не считывает и тогда выполняет ε-переход) и действует согласно одному из правил перехода. А именно, пусть автомат находится в состоянии q, на верхушке стека лежит символ $z \in \Gamma$ и автомат считывает букву σ. Тогда автомат выбирает одну из пар $(q', \gamma) \in \delta(q, \sigma, z)$, переходит в состояние q', снимает с верхушки символ z и добавляет в стек слово γ, причём, если $\gamma = \gamma_1 \gamma_2 \dots \gamma_n$, то γ_n оказывается снизу, а γ_1 сверху. Автомат завершает работу с ошибкой, если не может выполнить переход, а входное слово ещё не обработано.

Выделяют два типа магазинных автоматов, которые различаются по условию приёма входного слова. В первом случае, автомат P принимает слово w, если существует такая последовательность переходов, что в результате обработки слова, он оказался в принимающем состоянии, в этом случае автомат P является *допускающим по заключительному состоянию*. Во втором случае, автомат P принимает слово w, если существует такая последовательность переходов, что в результате[2] обработки слова, стек автомата оказался пуст, в этом случае автомат называется *допускающим по пустому магазину*.

Замечание 1. *Если* $\delta(q, \sigma, z) = \{(q_1, \gamma_1), (q_2, \gamma_2)\}$*, то автомат выбирает одну из пар и при переходе в состояние* q_1 *автомат помещает в стек* γ_1*, а при переходе в* q_2*, автомат помещает в стек* γ_2*. Автомат* ***не может*** *перейти в* q_1*, а в стек положить* γ_2*!*

Определение 2. *Конфигурацией* МП-автомата называется элемент множества $Q \times \Sigma^* \times \Gamma^*$. При начале работы на входе w автомат P находится в конфигурации (q_0, w, Z_0). За такт работы автомат изменяет конфигурацию, согласно правилам перехода. Если автомат находился в конфигурации $(q, \sigma v, Z_n \dots Z_2 Z_1 Z_0)$ и $\delta(q, \sigma, z) = \{(q_1, \gamma_1), (q_2, \gamma_2)\}$, то автомат либо переходит в конфигурацию $(q_1, v, \gamma_1 Z_{n-1} \dots Z_1 Z_0)$, либо в $(q_2, v, \gamma_2 Z_{n-1} \dots Z_1 Z_0)$. Верхушка стека в цепочке $\gamma \in \Gamma^*$ находится слева, дно стека находится справа.

На множестве конфигураций автомата P введено *отношение* $\underset{P}{\vdash}$, такое что если из конфигурации c_1 согласно функции перехода P есть переход в конфигурацию c_2, то $c_1 \underset{P}{\vdash} c_2$. Когда ясно о каком автомате идёт речь, мы будем опускать индекс отношения. Так, $(q, \sigma v, Z_n \dots Z_2 Z_1 Z_0) \vdash (q_1, v, \gamma_1 Z_{n-1} \dots Z_1 Z_0)$

Замечание 2. *При выполнении такта работы, автомат всегда снимает симол с верхушки стека. Например, если изначально автомат находился в конфигурации* (q_0, av, Z_0) *и перешёл в конфигурацию* (q, v, aZ_0)*, то правило, которое он применил выглядит как* $(q, aZ_0) \in \delta(q_0, a, Z_0)$*. Кстати, в отличие от грамматик, входной алфавит и алфавит стека могут пересекаться. То есть, условие* $\Sigma \cap \Gamma = \emptyset$ ***не*** *налагается.*

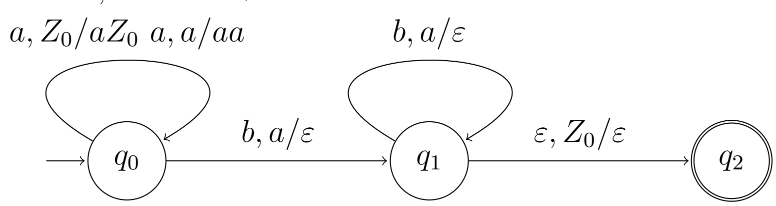
[2] под «в результате» понимается что после обработки слова w автомат оказался пуст. Если стек оказался пуст в процессе обработки слова, т.е. когда слово ещё не было прочитано до конца, то это не означает, что слово было принято автоматом

Напомним, что *транзитивным замыканием* бинарного отношения R называется минимальное транзитивное бинарное отношение R^*, содержащее R. То есть, если $(a, b) \in R$ и $(b, c) \in R$, то $(a, c) \in R^*$, даже если $(a, c) \notin R$. Кроме того, отношение R^* само по себе является транзитивным, то есть, если $(a, b) \in R^*$ и $(b, c) \in R^*$, то и $(a, c) \in R^*$.

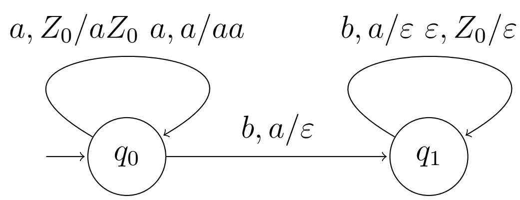
Определим условие приёма слова w автоматом P через транзитивное замыкание отношения $\vdash$. Если автомат P является допускающим по принимающему состоянию, то $w \in L(P)$ тогда и только тогда, когда $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \gamma)$, где $q \in F$, $\gamma \in \Gamma^*$. Если автомат P является допускающим по пустому магазину, то $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$, где $q \in Q$ – не обязательно принимающее состояние.

1.2 Примеры

Графически магазинные автоматы задаются следующим образом: для каждого правила $\delta(q, \sigma, Z) = (p, \gamma)$ на переходе из состояния q в состояние p пишут $\sigma, Z/\gamma$. Если автомат принимает слово по пустому магазину, то принято считать, что $F = \emptyset$.



Данный автомат является допускающим по завершающему состоянию.



Данный автомат является допускающим по пустому стеку.

Упражнение 1. Показать, что данные автоматы распознают язык $L = \{a^n b^n \mid n > 0\}$.

Определение 3. Магазинный автомат P является *детерминированным*, если множество $\delta(q, \sigma, Z)$ содержит не более одного правила $\forall \sigma \in \Sigma, \forall Z \in \Gamma$. Если для некоторой буквы σ, $\delta(q, \sigma, Z) \neq \emptyset$, то $\delta(q, \varepsilon, Z) = \emptyset$. То есть все переходы в автомате P определены однозначно и в случае когда из пары q, Z есть ε-переход, то других переходов из данной пары нет.

Упражнение 2. Показать, что автоматы, изображённые на диаграммах являются детерминированными.

Классическим примером КС-языков являются языки Дика. А именно, языком типа D_n будем называть язык состоящий из правильных скобочных выражений с n типами скобок. Формально, язык D_n определён над размеченым алфавитом $\Sigma = \Sigma_n \cup \bar{\Sigma}_n$ – в Σ_n входят открывающиеся скобки, в $\bar{\Sigma}_n$ закрывающиеся. Определим языки Дика индуктивно.

Определение 4. Язык Дика D_n задан грамматикой $S \to \sigma_i\bar{\sigma}_i \mid \sigma_i S\bar{\sigma}_i \mid SS$, где $i \in 1..n$.

Скобочным итогом $i-$го типа слова w, назовём число $\|w\|_i = |w|_{\sigma_i} - |w|_{\bar{\sigma}_i}$. Если w является првильным скобочным выражением, то для любого префикса $p : w = ps$ и любого $i \leqslant n$ справедливо $\|p\|_i \geqslant 0$ и $\|w\|_i = 0$. То есть,

$$w \in D_n \Rightarrow \forall i \leqslant n, \forall k \leqslant |w|, \|w[1,k]\|_i \geqslant 0, \|w\|_i = 0$$

Упражнение 3. Показать, что обратное неверно.

2 Задачи

1. Пусть D_2 – язык правильных скобочных выражений с двумя типами. Тогда $L = D_2 \cap ([_1|[_2)^*(_1]|_2])^*$. То есть, L – язык скобочных выражений ширины 1, то есть $[_1[_2[_1\,_1]_2]_1] \in L$, а $[_1[_2[_1\,_1]_2][_2\,_2]_1] \notin L$. Построить МП-автомат, распознающий язык L^*.

2. Привести алгоритм построения МП-автомата P, допускающего по заключительному состоянию по МП-автомату N, допускающего по пустому стеку. Привести алгоритм обратного построения по автомату P, автомата N. Если в задаче 1 Вы построили N-автомат, постройте по нему аналогичный P-автомат, если вы построили P-автомат, постройте по нему аналогичный N-автомат. Если вы не можете придумать алгоритм, его можно прочитать в книге Хопкрофта-Мотвани-Ульмана, но это не означает, что надо его перетехивать.

Задача 3. Построить КС-грамматику, порождающую

1) язык палиндромов $L \subseteq \{a, b\}^*$, то есть $L = \{w \mid w = w^R\}$. Например, палиндромами являются слова «ротор» и «казак».

2) язык $\bar{L}$.

4. Построить КС-грамматику G, порождающую L или МП-автомат M, распознающий L.

1. $L = \{a^i b^j c^k \mid i = j \vee i = k; i, j, k \geqslant 0\}$

2*. $L = \{w \mid w = uv \Rightarrow u \neq v\}$, то есть $w \in L$ непредставимо в виде uu.