Теория формальных языков

Тух Игорь и Ютман Михаил
по лекциям Михаила Эдуардовича Дворкина

19 марта 2018 г.

Содержание

1. Введение

См. Хопкрафт, Мотвани, Ульман. Введение в теорию автоматов, языков и вычислений

Эти типы стали есть на складе

***Определение* 1.1.** Зафиксируем конечное множество символов (алфавит) Σ и скажем, что слово $w_1 \dots w_n$ – это конечная последовательность символов из алфавита. Язык L – это множество слов (над алфавитом Σ).

Замечание. Язык L не обязан быть конечным.

Например, рассмотрим язык всех синтаксически корректных фраз на русском языке, тогда предложения будет словами, а кириллица, знаки препинания и пробелы – алфавитом. Заметим, что прапраправнук (сколько угодно раз пра) – корректное слово на русском. Значит, таких фраз бесконечное количество.

***Определение* 1.2.** Σ^* – это множество всех слов над Σ. ε – стандартное обозначение пустого слова.

Замечание. Множество всех слов над алфавитом – это счетное множество. Любой язык, как его подмножество, либо конечный, либо счетный (т.е. не более, чем счетный). Над пустым алфавитом есть два различных языка $\{\varepsilon\}$ и $\varnothing$.

Замечание. Пусть есть некоторая задача с ответом ДА или НЕТ (входные данные – слово над алфавитом). Тогда есть соответствие между некоторым языком и множеством слов, на которых ответ ДА. Задач с ответом ДА/НЕТ несчетное число. А различных программ на C++ счетное. Значит, существуют неразрешимые ДА/НЕТ задачи.

2. Детерминированные Конечные Автоматы (DFA)

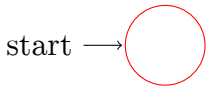
2.1. Детерминированные Конечные Автоматы

***Определение* 2.1.** Детерминированный конечный автомат $A = (\Sigma, Q, q_o, T, \delta)$, где

1. Σ – конечный алфавит;
2. Q – конечное множество состояний;
3. $q_o \in Q$ – начальное состояние;
4. $T \subseteq Q$ – множество терминальных состояний;
5. $\delta : Q \times \Sigma \to Q$ – функция переходов.

Обозначение.

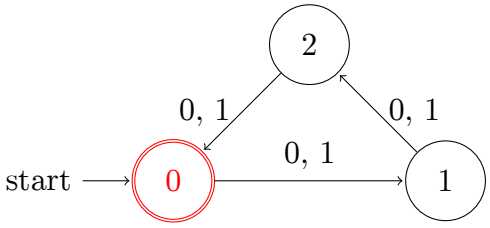
Далее в конспекте так будет обозначаться стартовые состояния:



А вот так терминальные:

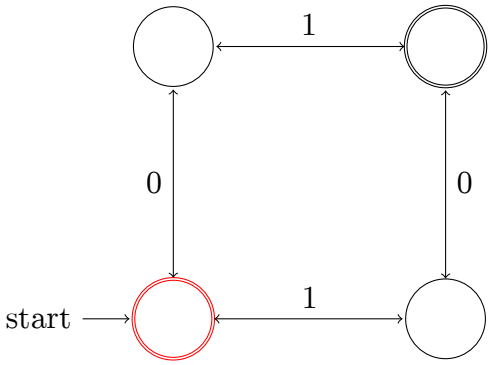
Примеры.

1. Зафиксируем алфавит $\{0, 1\}$. $L = \{w \mid |w| \vdots 3\}$



Переход по любому символу происходит по указанной стрелке. Значение в вершине совпадает с остатком по модулю 3 слова. Терминальное состояние совпадает с начальным.

2. $\Sigma = \{0, 1\}$



Как несложно понять, распознаются только слова четной длины.

***Определение* 2.2.** Обобщенная функция переходов $\hat{\delta} : Q \times \Sigma^* \to Q$, причем $\hat{\delta}(q, \epsilon) = q$ и $\hat{\delta}(q, xw) = \hat{\delta}(\hat{\delta}(q, x), w)$. И тогда язык принимаемый ДКА $L(A) = \{w \in \Sigma^* : \hat{\delta}(q_o, w) \in T\}$.

Замечание. Проверка принадлежности слова w языку $L(A)$ может быть осуществлена за $\mathcal{O}(|w|)$.

2.2. Минимизация ДКА

Задача минимизации: $A_1 \to A_2$ так, что $L(A_1) = L(A_2)$ и $|Q(A_2)| \to \min$.

***Определение* 2.3.** Состояния p и q – различимые, если $\exists w \in \Sigma^* \;\; \hat{\delta}(p, w) \in T \oplus \hat{\delta}(q, w)$.

Алгоритм (поиска всех пар различимых состояний)**.**

1. $p \in T, q \notin T \Rightarrow (p, q)$ – различима;
2. $\hat{\delta}(p, x) = u$ и $\hat{\delta}(q, x) = v$. Тогда (u, v) различима $\Rightarrow (p, q)$ – различима.

Нужно запустить DFS или BFS из вершин из первого пункта. Доказательство корректности очевидно по индукции. Время работы будет $\mathcal{O}(|Q|^2)$ (ребер вообще говоря, $|Q|^2 \cdot |\Sigma|$).

Лемма. "p неразличима с q" является отношением эквивалентности.

Доказательство. Рефлексивность, симметричность и транзитивность очевидны. □

Алгоритм (Минимизации ДКА)**.**

Рассмотрим класс эквивалентности. Будем строить автомат A_2. Новые состояния соответствуют классам эквивалентности. Начальным состоянием будет класс начального состояния. Каждый класс либо состоит только из терминальных, либо только из нетерминальных, на основании чего определяем терминальность в новом автомате. Несложно заметить, что не существует двух переходов по одной и той же букве из двух состояний из одного класса эквивалентности в два разных класса (тогда эти две вершины были бы различимы). Оставляем только состояния, достижимые из начального. Получили автомат, принимающий те же слова. Автомат меньшего размера получить нельзя, так как все классы эквивалентности нужны.

Замечание. Можно легко обобщить понятие различимости на вершины двух разных автоматов (пара состоит из одной вершины из одного автомата и еще одной из другой). Тогда аналогично найдем все различимые пары вершины в этих двух автоматах. Проверка на эквивалентность этих двух автоматов заключается в проверке различимости стартовых состояний.

2.3. Правые контексты

***Определение* 2.4.** Правым контекстом называется функция $C_L^R : \Sigma^* \to 2^{\Sigma^*}$, $C_L^R(w) = \{u \in \Sigma^* : wu \in L\}$

Аналогично можно определить левые контексты.

Замечание. $w \in L \Leftrightarrow \epsilon \in C_L^R(w)$

Замечание. $\forall w \in \Sigma^* \;\; C_L^R(w)$ – различные множества правых контекстов. Количество $= |Q_{minA}|$.

***Определение* 2.5.** Тупиковым (дьявольским) состоянием ДКА называется состояние, все переходы из которого ведут в него же самого.

Замечание. Тупиковые состояния на картинке обычно не рисуют.

3. Недетерминированные конечные автоматы с ε-переходами

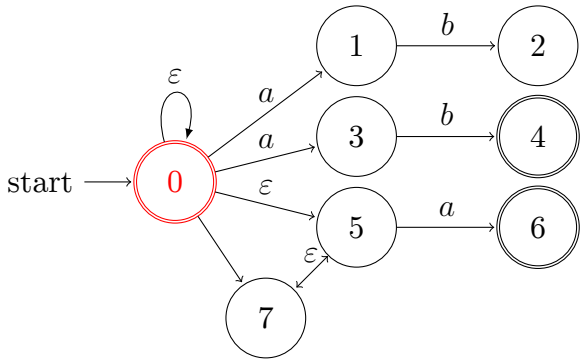
3.1. Недетерминированные конечные автоматы с ε-переходами

***Определение* 3.1.** Недетерминированный конечный автомат (НКА) – это пятёрка (Σ, Q, q_0, T, δ), где Q – конечное, $q_0 \in Q, T \subseteq Q, \delta : Q \times (\{\varepsilon\} \cup \Sigma) \to 2^Q$, которая принимает строку w и выдаёт "Да", если есть хотя бы один путь по строке w из начальной вершины в терминальную.

***Определение* 3.2.** ε-переход – это переход, который можно спонтанно сделать в любой момент чтения слова.

Число переходов, возможно, экспоненциально больше, чем у детерминированного, но тем не менее конечно.

Пример. Рассмотрим следующий автомат:



В этот автомат слово "a" можно принять перейдя из 0 в 3, а можно перейти по ε-переходу из 0 в 5, потом походить миллион раз туда-обратно по ε-переходу между 5 и 7, а потом из 5 перейти в 6, и таким образом, это слово распознаётся автоматом.

***Определение* 3.3.** Язык, распознаваемый НКА, $L(A) = \{w \in \Sigma^* : \exists$ путь из q_0 в $q \in T$, на ребрах которого написано слово $w\}$.

Пример. **TODO:** картинка

***Определение* 3.4.** ε-замыкание вершины q edge это множество состояний, достижимых из q по ε-переходам (можно закодить dfs-ом).

Каждый раз добавляем вместо вершины её ε-замыкание. Берём ε-замыкания и их битовые маски or-им. Мы договорились, что у нас размер алфавита константа, поэтому для каждого состояния мы могли предподсчитать битовую маску. Поэтому, если мы всё предподсчитали, то у нас переход по очередной букве работает за $O(|Q|^2)$, но такая оценка почти никогда не достигается, поэтому на деле алгоритм работает примерно за $|w|$ (но это не точно).

Если мы посмотрим на все возмножные замыкание, получится, что есть $2^{|}Q|$ различных. Таким образом, если программа хранит всегда конечное число информации и читает слово слева направо, то она является детерминированным конечным автоматом. Получлось, что программа, которая осуществляет переходы по недетерминированному конечному автомату является детерминированным конечным автоматом.

3.2. Детерминизация ε-НКА

Задача сделать из ε-НКА A_1 ДКА A_2, такой что $L(A_2) = L(A_1)$.

$A_2 = (\Sigma, 2^{Q_{A_1}}, \varepsilon\text{-closure}(q_{0A_1}), s : s \cap T \neq \varnothing, \delta(S, a) = \bigcup\limits_{p \in S, p \to q} \varepsilon\text{-closure}(q))$.

Пример. $L_1 = \{w : w_n = 0\}$

TODO: картинка

Пример. $L = \{w : w_{|w|-n+1} = 0\} = L_1^R$. ε-НКА:

TODO: картинка

В ДКА для того же языка $|Q| \geq 2^n$ (оцените разницу по сравнению с $n+1$ цифрой в ε-НКА).

В НКА каждый переход в худшем случае за $|Q|^2$, а в ДКА – это просто переход в прямоугольном массиве.

Идея: попробуем детерминизировать НКА, и если за мало операций получилось детерминизировать, то будем работать с ДКА, а если не повезло, то будем работать с НКА.

3.3. Произведение конечных автоматов

Пусть есть два автомата: A и B, тогда $A \times B = (\Sigma, Q_A \times Q_B, (q_{0_A}, q_{0_B}), \ldots, \delta((p, q), a) = (\delta_A(p, a), \delta_B(q, a)))$.

Если $T = T_A \times T_B$, то это $L_1 \cap L_2$, если же $T = T_A \times Q_B \cup Q_A \times T_B$, то это $L_1 \cup L_2$.