Лекция 3. Недетерминированные конечные автоматы

Главные вопросы, которые мы обсуждаем, представлены на СЛАЙДЕ 1. Введем недетерминизм в поведение КА, включая переходы из одного состояния в другое по пустой строке.

3.1 Недетерминированные конечные автоматы

Недетерминированный конечный автомат (NFA – *Nondeterministic Finite Automaton*) обладает свойством находиться в нескольких состояниях одновременно. Эту особенность часто представляют как свойство автомата догадываться относительно входных данных. Скажем, если НКА используется для поиска всех идентификаторов в программе (текстовой строке большой длины), то в начале поиска крайне желательно «догадаться» о том, что НКА находится в начале одной из требуемых цепочек, а затем использовать некоторую последовательность состояний для проверки того, что знак за знаком появляется данная цепочка.

Далее мы формально определим НКА и покажем, что такие автоматы допускают регулярные языки, также как и ДКА. Интересно отметить, что обычно НКА компактнее ДКА и легче строятся. Более того, несмотря на существование возможности преобразования НКА в ДКА, последние могут иметь экспоненциально больше состояний, чем НКА.

3.2 Неформальное описание НКА

Итак, НКА имеют конечные множества состояний и входных символов, одно начальное состояние и множество заключительных состояний, а также функцию переходов. Именно в ней, собственно говоря, и заключается различие между ДКА и НКА. Аргументы те же, а вот результаты различные. Как мы видели ранее, у ДКА это одно состояние, а у НКА – множество, в том числе пустое. Далее мы дадим строгое определение НКА, но сначала рассмотрим пример.

Пример 13.

На СЛАЙДЕ 2 приведен НКА, допускающий те и только те строки, которые заканчиваются на 01. Начальным состоянием является q_0, и в этом состоянии он будет находиться до тех пор, пока не догадается, что началась замыкающая подстрока 01. Всегда существует вероятность, что следующий символ не является начальным для интересующей нас подстроки, даже если это символ 0. Значит, состояние q_0 может иметь переходы в само себя как по 1, так и по 0.

Если очередной входной символ – 0, то НКА может «подумать», что уже началась замыкающая подстрока 01. Тогда дуга с меткой 0 ведет из q_0 в q_1. Но из начального состояния мы имеем две дуги, помеченные символом 0, иными словами НКА может перейти и в q_0 в q_1. В том, что он туда действительно переходит, мы убедимся при обсуждении формального определения НКА. В состоянии q_1 проверяется равенство следующего символа 1. Если это так, то он переходит в состояние q_2.

Из состояния q_1 нет дуги, помеченной 0, а у состояния q_2 вообще нет исходящих дуг. Считается, что в этих состояниях пути НКА прерываются, хотя другие пути могут по-прежнему существовать. Мы видели выше, что ДКА имеет в каждом состоянии ровно одну исходящую дугу для каждого входного символа. Так вот для НКА нет такого ограничения. В примере 13 мы убедились, что дуг может быть 0, 1 или 2.

После запуска «симуляции» нашего НКА мы увидим, что происходит, когда автомат получает на вход последовательность 00101. Работа, как обычно, начинается в состоянии q_0. Когда прочитан первый 0, НКА может перейти в q_0 либо q_1, поэтому он переходит в оба состояния.

Затем считывается второй 0. Из q_0 вновь можно перейти в q_0 либо q_1. Однако в q_1 нет переходов по символу 0, поэтому оно «*умирает*» (или «*попадает в тупик*»). Когда появляется третий входной символ, т.е. 1, НКА должен рассматривать переходы из q_0 и q_1. Из q_0 по 1 возможен переход только в q_0, а из q_1 – в q_2. Прочитав подстроку 001, НКА находится в двух состояниях – q_0 и q_2. Поскольку q_2 заключительное состояние, то НКА допускает данную подстроку.

К сожалению, во входной строке есть еще символы. Четвертый входной символ, т.е. 0, приводит к отмиранию ветви q_2, а q_0 переходит в q_0 и q_1. По пятому (последнему) символу 1 из q_0 происходит переход в q_0, а из q_1 в q_2. НКА попал в заключительное состояние, а значит, строка 00101 допустима. См. также схему работы нашего НКА на СЛАЙДЕ 3.

3.3 Формальное определение НКА

Структура НКА в целом повторяет структуру ДКА (СЛАЙД 4).

$A = (Q, \Sigma, \delta, q_0, F)$.

где Q – конечное множество состояний, Σ – конечное множество входных символов, δ – функция переходов, q_0 – начальное состояние ($q_0 \in Q$), F – множество заключительных (допускающих) состояний ($F \subseteq Q$).

Функция переходов – это функция, аргументами которой являются состояние из Q и входной символ из Σ. Значением этой функции является множество состояний, в которых может оказаться НКА.

Пример 14.

Наш НКА из примера 13 можно формально задать так:

$A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$, где функция переходов задается таблично. СЛАЙД 5.

Функцию переходов можно задавать таблично или графически. В первом случае единственное отличие от аналогичных таблиц для ДКА заключается в том, что для НКА на пересечениях строк и столбцов стоят множества, причем некоторые из них могут быть **одноэлементными** или синглетонами (*singleton*). Когда из некоторого состояния по определенному символу нет перехода, то на пересечении строки и столбца должно стоять пустое множество.

В дальнейшем нам понадобится расширенная функция переходов (РФП) НКА. Аргументами функции $\hat{\delta}$ являются состояние q и цепочка входных символов w, значением – множество состояний, в которые НКА попадает из состояния q, обработав строку w. В случае использования НКА из примера 13 $\hat{\delta}(q, w)$ – это столбец состояний, которые получаются при чтении строки w, при условии, что q – единственное состояние в первом столбце. Например, $\hat{\delta}(q_0, 001) = \{q_0, q_2\}$. СЛАЙД 6.

Формально РФП определяется следующим образом (СЛАЙД 7).

1. $\hat{\delta}(q, \varepsilon) = \{q\}$

2. Допустим, строка w имеет вид $w = xa$, где a – последний символ строки, а x – все

остальное, и предположим, что $\hat{\delta}(q,x)=\{p_1,p_2,\ldots,p_k\}$. Пусть

$$\bigcup_{i=1}^{k}\delta(p_i,a)=\{r_1,r_2,\ldots,r_{\mathrm{m}}\}.$$

Тогда $\hat{\delta}(q,w)=\{r_1,r_2,\ldots,r_m\}$, т.е. чтобы найти $\hat{\delta}(q,w)$, нужно найти $\hat{\delta}(q,x)$, а затем из всех полученных состояний совершить переходы по символу a.

Пример 15.

Используем расширенную функцию переходов для описания того, как НКА из примера 13 обрабатывает строку 00101. СЛАЙД 8.

1. $\hat{\delta}(q_0,\varepsilon)=\{q_0\}$.
2. $\hat{\delta}(q_0,0)=\delta(q_0,0)=\{q_0,q_1\}$.
3. $\hat{\delta}(q_0,00)=\delta(q_0,0)\cup\delta(q_1,0)=\{q_0,q_1\}\cup\varnothing=\{q_0,q_1\}$.
4. $\hat{\delta}(q_0,001)=\delta(q_0,1)\cup\delta(q_1,1)=\{q_0\}\cup\{q_2\}=\{q_0,q_2\}$.
5. $\hat{\delta}(q_0,0010)=\delta(q_0,0)\cup\delta(q_2,0)=\{q_0,q_1\}\cup\varnothing=\{q_0,q_1\}$.
6. $\hat{\delta}(q_0,00101)=\delta(q_0,1)\cup\delta(q_1,1)=\{q_0\}\cup\{q_2\}=\{q_0,q_2\}$.

3.4 Язык НКА

Внимательный слушатель (и читатель) должен был заметить, что по приведенному описанию НКА он допускает строку w, если при ее прочитывании автомат может выбрать хотя бы одну последовательность переходов в следующие состояния так, чтобы перейти в одно из заключительных состояний из начального. Тот факт, что при выборе некоторой последовательности переходов по символам входной строки мы можем попасть в незаключительное состояние и даже вообще не попасть ни в какое (последовательность «умирает»), совсем не означает, что входная строка w не является допустимой для всего НКА.

Формально, если $A=(Q,\Sigma,\delta,q_0,F)$ – некий НКА, то

$$L(A)=\left\{w\middle|\hat{\delta}(q_0,w)\cap F=\varnothing\right\}.$$

Тогда, $L(A)$ – это множество строк w из $\boldsymbol{\Sigma}^*$, для которых среди состояний $\hat{\delta}(q_0,w)$ есть по крайней мере одно заключительное. СЛАЙД 9.

Пример 16.

Докажем, что НКА из примера 13 допускает язык L = {w | w оканчивается на 01}. Для этого мы докажем три следующих утверждения (СЛАЙД 10):

1. $\hat{\delta}(q_0,w)$ содержит q_0 для любой строки w.
2. $\hat{\delta}(q_0,w)$ содержит q_1 тогда и только тогда, когда w заканчивается на 0.
3. $\hat{\delta}(q_0,w)$ содержит q_2 тогда и только тогда, когда w заканчивается на 01.

Чтобы их доказать, нужно рассмотреть, как A может попасть в каждое из этих состояний, т.е. каким был последний входной символ, и в каком он находился в состоянии перед чтением этого символа. Поскольку язык этого НКА – это множество строк w, для которых $\hat{\delta}(q_0,w)$ содержит единственное заключительное состояние q_2, то доказательство этих трех утверждений (а по большому счету достаточно доказать только третье) гарантирует, что язык данного НКА есть набор строк, оканчивающихся на 01. Доказательство представляет собой индукцию по длине строки, начиная с нулевой.

Итак, если $|w|$ = 0, то $w = \varepsilon$. В первом утверждении говорится, что $\hat{\delta}(q_0,\varepsilon)$ содержит

q_0. Это так в силу базисной части этого утверждения относительно РФП. Далее, мы знаем, что ε не заканчивается на 0, и $\hat{\delta}(q_0, \varepsilon)$ не содержит q_1. Получаем, что гипотезы «тогда и только тогда» во втором утверждении являются ложными в обе стороны. Следовательно, само утверждение является истинным в обе стороны. Аналогично проводится доказательство третьего утверждения.

Теперь допустим, что $w = xa$, где a – это символ 0 или 1. Предположим, что три наших утверждения выполняются и для x. Нужно их доказать для w, предположив что $|w| = n + 1$, а $|x| = n$. Допустив верность гипотезы для n, докажем ее для $n + 1$.

1. Мы знаем, что $\hat{\delta}(q_0, x)$ содержит q_0. По обоим входным символам существуют переходы из q_0 в себя, то $\hat{\delta}(q_0, w)$ также содержит q_0. Следовательно, утверждение 1 доказано для w.

2. Пусть w оканчивается на 0, т.е. $a = 0$. Применяя утверждение (1) к x, мы получим, что $\hat{\delta}(q_0, x)$ содержит q_0. Но существует переход из q_0 в q_1 по символу 0. Отсюда – $\hat{\delta}(q_0, w)$ содержит q_1.

Пусть $\hat{\delta}(q_0, w)$ содержит q_1. Из диаграммы переходов нашего НКА видно, что в состояние q_1 мы можем попасть только одним способом, когда $w = x0$. Следовательно, мы доказали достаточность и необходимость в утверждении (2).

3. Пусть w оканчивается на 01. Тогда если $w = xa$, то известно, что $a = 1$, а x оканчивается на 0. Применяя утверждение (2) к x, мы получим, что $\hat{\delta}(q_0, x)$ содержит q_1. Но существует переход из q_1 в q_2 по символу 1. Отсюда – $\hat{\delta}(q_0, w)$ содержит q_2.

Пусть $\hat{\delta}(q_0, w)$ содержит q_2. Из диаграммы переходов нашего НКА видно, что в состояние q_2 мы можем попасть только одним способом, когда $w = x1$ и $\hat{\delta}(q_0, x)$ содержит q_1. Применяя утверждение (2) к x, мы получим, что x оканчивается на 0, а w – на 01. Следовательно, мы доказали достаточность и необходимость в утверждении (3).

3.5 Эквивалентность ДКА и НКА

Для языка строк, оканчивающихся на 0 и 1 (и для многих других, конечно), создать НКА немного легче, чем ДКА. Несмотря на это, любой язык, описываемый некоторым НКА, можно описать и ДКА. Обычно детерминированный КА имеет столько же состояний и гораздо большее число переходов. Наихудший случай: ДКА может содержать 2^n состояний, а НКА для того же языка имеет n состояний.

Когда пытаются доказать, что у ДКА есть все возможности НКА, используют **конструкцию подмножеств**, т.к. она включает построение всех подмножеств множества состояний НКА. Для нас это еще и хороший пример того, как один КА описывается в терминах состояний и переходов другого КА без подробных сведений об этом «другом». СЛАЙД 11.

Немного изменив использовавшиеся выше обозначения, попробуем построить подмножества. Имеем НКА $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$, и нам нужно описать ДКА $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$, у которого $L(N) = L(D)$. Входные алфавиты у обоих автоматов совпадают, а начальное состояние D – это множество, которое содержит только начальное состояние автомата N.

Q_D – это булеан множества Q_N. Если в Q_N содержатся n состояний, то в Q_D их будет уже 2^n. Во многих случаях не все они достижимы из начального состояния автомата D. Такие недостижимые состояния можно устранить, поэтому фактическое число состояний автомата D может оказаться меньше, чем 2^n.

F_D – это множество подмножеств S множества Q_N, для которых $S \cap F_N \neq \varnothing$. Проще говоря, оно состоит из всех множеств состояний N, содержащих по крайней мере одно заключительное состояние.

Для любого множества $S \subseteq Q_N$ и любого входного символа a из Σ функция переходов $\delta_D(S,a) = \bigcup_{p \in S} \delta_N(p,a)$. Для того, чтобы найти функцию переходов ДКА, просматриваются все состояния p из множества S, и осуществляется поиск тех состояний N, в которые можно попасть из p по символу a. Затем множества найденных состояний объединяются по всем состояниям p. СЛАЙД 12.

Пример 17.

Пусть N – это НКА их примера 13. Мы знаем, что множество состояний есть $\{q_0, q_1, q_2\}$, в нем три элемента. Значит, конструкция подмножеств дает ДКА с 8 состояниями, отвечающими всем подмножествам, составленным из этих трех состояний. На СЛАЙДЕ 13 приведена следующая таблица переходов для полученных 8 состояний.

Данная таблица соответствует ДКА, т.к. его состояния **сами являются множествами**. Можно переобозначить состояния. Пустое множество обозначить как A, $\{q_0\}$ – как B, $\{q_1\}$ – как C и так далее. На СЛАЙДЕ 14 показана та же таблица переходов, но с переобозначенными состояниями.

Очевидно, что начиная в состоянии B, из всех 8 состояний можно попасть только в B, E, F. Остальные 5 состояний из начального состояния недостижимы, а значит, их можно из таблицы удалить. Как правило, можно избежать построения элементов таблицы переходов для всех подмножеств, что требует экспоненциального времени. Для этого выполняется следующее вычисление подмножеств.

Синглетон из начального состояния N является достижимым. Пусть мы установили, что множество состояний S является достижимым. Тогда для каждого входного символа a нужно найти множество состояний $\delta_D(S,a)$. Найденные множества состояний также будут достижимы (СЛАЙД 15).

Итак, нам известно, что $\{q_0\}$ является одним из состояний ДКА. Находим, что $\delta_D(\{q_0\},0) = \{q_0, q_1\}$ и $\delta_D(\{q_0\},1) = \{q_0\}$. Так была получена вторая строка таблицы.

Одно из найденных множеств, а именно $\{q_0\}$ уже рассматривалось, однако второе – $\{q_0, q_1\}$ новое, и для него нужно найти переходы. Получаем, $\delta_D(\{q_0, q_1\},0) = \{q_0, q_1\}$ и $\delta_D(\{q_0, q_1\},1) = \{q_0, q_2\}$. Последнее выражение мы, в частности, получили следующим образом: $\delta_D(\{q_0, q_1\},1) = \delta_N(q_0,1) \cup \delta_N(q_1,1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$. В итоге имеем 5-ую строку таблицы и «новоиспеченное» состояние $\{q_0, q_2\}$.

Аналогично для шестой строки таблицы:

$$\delta_D(\{q_0, q_2\},0) = \delta_N(q_0,0) \cup \delta_N(q_2,0) = \{q_0, q_1\} \cup \varnothing = \{q_0, q_1\},$$

$$\delta_D(\{q_0, q_2\},1) = \delta_N(q_0,1) \cup \delta_N(q_2,1) = \{q_0\} \cup \varnothing = \{q_0\}.$$

И что самое замечательное: мы не получили ни одного нового множества состояний, а конструкция множества сошлась. Известны все допустимые состояния и соответствующие им переходы. Полученный ДКА показан на СЛАЙДЕ 16. По случайному стечению обстоятельств у него, как и у НКА, оказалось три состояния. Однако, количество переходов у ДКА – 6, а у НКА – 4.

Теперь мы приблизились к доказательству эквивалентности языков и автоматов (СЛАЙДЫ 17-19).

Теорема 3.1. Если ДКА $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ построен по НКА $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ посредством конструкции подмножеств, то $L(D) = L(N)$.

Доказательство. Сначала покажем, что $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(\{q_0\}, w)$, попутно заметив, что и для обеих расширенных функций переходов значением является множество состояний из Q_N. При этом первая функция интерпретирует его как состояние из Q_D, являющегося булеаном Q_N, а вторая – как подмножество Q_N.

Пусть $w = \varepsilon$. Из определений расширенных функций переходов ДКА и НКА имеем $\hat{\delta}_D(\{q_0\}, \varepsilon) = \{q_0\}$ и $\hat{\delta}_N(\{q_0\}, \varepsilon) = \{q_0\}$.

Теперь пусть $|w| = n + 1$, и наше утверждение верно для n. Как уже делали не раз, разбиваем w на две части, получая $w = xa$. Согласно нашей гипотезе, $\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(\{q_0\}, x)$. Допустим, множества состояний НКА N представляют собой $\{p_1, p_2, \ldots, p_k\}$. Согласно определению НКА:

(1) $\hat{\delta}_N(\{q_0\}, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$.

Со своей стороны, конструкция подмножеств дает для ДКА следующее:

(2) $\hat{\delta}_D(\{p_1, p_2, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$.

Подставляя (2) в определение ДКА и используя $\hat{\delta}_D(\{q_0\}, w) = \{p_1, p_2, \ldots, p_k\}$, получаем:

(3) $\hat{\delta}_D(\{q_0\}, w) = \delta_D\left(\hat{\delta}_D(q_0, x), a\right) = \delta_D(\{p_1, p_2, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$.

Из (1) и (3) видно, что $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$. Заметим, что как D, так и N допускают строку w тогда и только тогда, когда соответственно $\hat{\delta}_D(\{q_0\}, w)$ или $\hat{\delta}_N(q_0, w)$ содержат некоторое заключительное состояние из F_N, а значит, мы получили полное доказательство $L(D) = L(N)$.

Теорема 3.2. Язык L допускается некоторым ДКА, если он допускается некоторым НКА.

Доказательство. Достаточность этого утверждения следует из конструкции подмножеств и также теоремы 3.1 (СЛАЙДЫ 20-21).

Доказательство необходимости также не представляет большой сложности. Неформально мы можем использовать переход от ДКА к идентичному НКА, причем диаграмму переходов для некоторого ДКА можно рассматривать как диаграмму для НКА, у которого по одному символу есть один переход. У нас есть ДКА $D = (Q, \Sigma, \delta_D, q_0, F)$, определим НКА $N = (Q, \Sigma, \delta_N, q_0, F)$ как автомат, эквивалентный D, где функция переходов определена правилом: если $\delta_D(q, a) = p$, то $\delta_N(q, a) = p$. Мы не будем приводить доказательство того, что если $\hat{\delta}_D(q_0, w) = p$, то $\hat{\delta}_N(q_0, w) = \{p\}$. Оно достаточно широко представлено в специальной литературе. Следствием является утверждением, что D допускает w тогда и только тогда, когда N допускает w, т.е. $L(D) = L(N)$

3.6 Конечные автоматы с эпсилон-переходами

Еще одним обобщением понятия КА являются КА с пустыми переходами. Иначе говоря, у автомата появляется новое свойство – возможность совершать переходы по пустой строке (или **спонтанно**), не получая на вход никакого символа. Класс языков, допустимых КА, это не расширяет, но придает больше удобства для программной реализации. В следующем разделе лекционного курса будут обсуждаться регулярные выражения, которые тесно связны с КА с эпсилон-переходами (далее мы их будем называть **ε-НКА**). Кроме того, они полезны при доказательстве эквивалентности между некоторыми классами языков.

3.7 Использование ε-переходов

По сложившейся традиции мы сначала неформально опишем ε-НКА, а затем дадим его формальное определение. Для неформального описания будем использовать диаграммы переходов с ε в качестве возможной метки. Автомат, в таком случае, можно рассматривать как допускающий последовательность меток, среди которых могут встречаться ε, вдоль путей из начального состояния в заключительное. При этом каждая пустая строка «невидима», не добавляя ничего в последовательность. СЛАЙД 22.

Пример 18.

На СЛАЙДЕ 23 показан ε-НКА, который допускает десятичные вещественные числа, состоящие из следующих элементов:

1) Необязательный знак.

2) Цепочка десятичных цифр, возможно пустая, если цепочка из п.4 не пуста.

3) Десятичная точка.

4) Цепочка десятичных цифр, возможно пустая, если цепочка из п.2 не пуста.

Обратим внимание на переход из q_0 в q_1 по любому из заданных символов. В этом случае q_1 – это такое состояние, когда не прочитана ни одна из цифр, ни десятичная точка, q_2 – это такое состояние, когда прочитана десятичная точка, а цифры целой части уже были прочитаны полностью, либо нет. В q_4 прочитана хотя бы одна цифра, но не прочитана точка. Как следствие, q_3 интерпретируется как ситуация, когда прочитана точка и хотя бы одна цифра перед ней или после нее. В этом состоянии КА может оставаться, продолжая считывать цифры, но может и спонтанно перейти в заключительное состояние q_5, когда цепочка цифр закончилась.

Пример 19.

На СЛАЙДЕ 24 показан ε-НКА, который распознает слова *red* и *edit*. Для каждого слова строится полная последовательность состояний, как если бы это было единственное слово для распознавания автоматом. Затем добавляется новое начальное состояние с ε-переходами в начальные состояния КА для каждого из ключевых слов.

3.8 Формальная запись ε-НКА

ε-НКА представляется точно так же, как НКА, с той разницей, что функция переходов должна содержать информацию о переходах по пустой цепочке. В формальной записи ε-НКА только функция переходов отличает его от НКА. Аргументами этой функции являются состояние из множества Q и элемент множества $\Sigma \cup \{\varepsilon\}$ – либо входной символ, либо пустая строка. СЛАЙД 25.

Пример 20.

На СЛАЙДЕ 26 приведена таблица переходов для ε-НКА из примера 18. Сам КА формально представляется так:

$$E = (\{q_0, q_1, \ldots, q_5\}, \{.,+,-,0,1,\ldots,9\}, \delta, q_0, \{q_5\}).$$

ε-замыкание

Теперь перейдем к формальному определению расширенной функции переходов для ε-НКА, которое позволит определиться с допустимостью строк и языков для таких КА. Перед этим дадим определение важного понятия ε-замыкания состояния. Неформально оно получается при совершении всех возможных переходов, помеченных ε, из некоторого состояния q. После совершения этих переходов в новые состояния снова могут выполняться переходы, помеченные символом ε. В конце концов, находятся все состояния, в которые можно попасть из q по любому пути, каждый переход в котором отмечен ε.

Формально ε-замыкание *ECLOSE* определяется рекурсивно следующим образом. *ECLOSE*(q) содержит q. Если *ECLOSE*(p) содержит p, и есть переход из в p в r, помеченный ε, то *ECLOSE*(q) содержит r. Более точно, если δ есть функция переходов ε-НКА, и *ECLOSE*(q) содержит p, то *ECLOSE*(q) содержит также все состояния $\delta(p, \varepsilon)$.

СЛАЙД 27.

Пример 21.

У автомата из примера 18 каждое состояние является собственным ε-замыканием, за исключением $ECLOSE(q_0) = \{q_0, q_1\}$ и $ECLOSE(q_3) = \{q_3, q_5\}$. Это связано с тем, что в данном КА всего два ε-перехода. Один из них добавляет q_1 в замыкания для q_0, а второй – q_5 в замыкание для q_3. СЛАЙД 28.

Пример 22.

На СЛАЙДЕ 29 приведен фрагмент ε-НКА. В случае надобности вычисления $ECLOSE(1)$ мы легко можем это сделать. $ECLOSE(1) = \{1, 2, 3, 4, 6\}$. В каждое состояние из этого множества можно попасть, проследив путь, помеченный только лишь ε. В 6-ое состояние КА приходит по пути 1->2->3->6. В 7-ое состояние он может также перейти из 1-го состояния, однако на пути встречается переход 4->5, не отмеченный символом ε. В 6-ое состояние есть второй путь (1->4->5->6), который при этом содержит не ε-переход. Для формирования замыкания достаточно одного пути, упомянутого выше.

Расширенные переходы и языки, допускаемые ε-НКА

С помощью замыкания легко выяснить переходы ε-НКА для заданной последовательности входных символов, а далее определить что означает для данного КА допустимость входной строки.

Предположим у нас есть ε-НКА $E = (Q, \Sigma, \delta, q_0, F)$. Для отображения процесса, сопровождающего чтение элементов заданной строки, нужно определить расширенную функцию переходов для состояния q и строки w через множество состояний, куда КА может попасть по путям, конкатенации меток, вдоль которых дают строку w. Естественно, символы ε ничего к w не добавляют. Получаем следующее определение (СЛАЙД 30):

(1) $\hat{\delta}(q, \varepsilon) = ECLOSE(q)$.

(2) Пусть наша строка $w = xa$, где a – последний символ w. Символ a не может быть, по понятным причинам, ε. Тогда определяем расширенную функцию переходов следующим образом.

(3) $\hat{\delta}(q, x) = \{p_1, p_2, \ldots, p_k\}$, где p_i – это те и только те состояния, в которые можно попасть из q по пути, отмеченному символом x. Путь может содержать либо оканчиваться одним или несколькими ε-переходами.

(4) $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \ldots, r_m\}$. Нужно совершить все переходы, помеченные a, из тех состояний, в которые КА может попасть из q по пути с меткой x. Состояния же r_i – не все, а только **некоторые** из тех, куда КА может попасть из q по пути с меткой w. В остальные состояния можно попасть из состояний r_i по переходам с меткой ε, как указано далее в (5).

(5) $\hat{\delta}(q, w) = \bigcup_{j=1}^{m} ECLOSE(r_j)$. Здесь как раз учитывается возможность существования дополнительных дуг, отмеченных ε, переход по которым возможен после перехода по последнему символу $a \neq \varepsilon$.

Пример 23.

Попробуем определить расширенную функцию переходов $\hat{\delta}(q_0, 6.7)$ для ε-НКА из примера 18. Для этого должны быть выполнены следующие шаги (СЛАЙД 31).

(0) $\hat{\delta}(q_0, \varepsilon) = ECLOSE(q_0) = \{q_0, q_1\}$.

(1) $\delta(q_0, 6) \cup \delta(q_1, 6) = \{q_1, q_4\}$.

(2) $\hat{\delta}(q_0, 6) = ECLOSE(q_1) \cup ECLOSE(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$.

(3) $\delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$.

(4) $\hat{\delta}(q_0, 6.) = ECLOSE(q_2) \cup ECLOSE(q_3) = \{q_2\} \cup \{q_3, q_5\} = \{q_2, q_3, q_5\}$.

(5) $\delta(q_2, 7) \cup \delta(q_2, 7.) \cup \delta(q_5, 7.) = \{q_3\} \cup \{q_3\} \cap \varnothing = \{q_3\}$.

(6) $\hat{\delta}(q_0, 6.7) = ECLOSE(q_3) = \{q_3, q_5\} = \{q_3, q_5\}$. Что от нас и требовалось.

Язык ε-НКА с наработанным нами опытом определяется просто. Если E – ε-НКА, то $L(E) = \left\{w \middle| \hat{\delta}(q, w) \bigcap F \neq \varnothing\right\}$. Очевидно, что язык – это множество строк, переводящих КА из начального состояния в одно (возможно и больше) из заключительных состояний. В примере 23 расширенная функция переходов содержала состояние q_5, которое является заключительным. Следовательно, строка 6.7 принадлежит языку, допускаемому ε-НКА. СЛАЙД 32.

Устранение ε-переходов

Для любого ε-НКА E можно найти ДКА D, допускающий тот же язык. Поскольку состояния D являются подмножествами из состояний E, то мы будем использовать механизм, похожий на уже знакомую нам конструкцию подмножеств, с применением замыканий.

Пусть $E = (Q_E, \Sigma, \delta_E, q_E, F_E)$. Определим эквивалентный ему ДКА $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$. СЛАЙДЫ 33 и 34.

(1) Здесь Q_D есть множество подмножеств Q_E. Для D допустимыми состояниями являются только **ε-замкнутые подмножества** Q_E. Иными словами, $S \subseteq Q_E, где\ S = ECLOSE(S)$. По-другому, ε-замкнутые подмножества состояний S – это такие множества, у которых любой ε-переход из состояния, принадлежащего S, приводит в состояние из S.

(2) $Q_D = ECLOSE(q_0)$. Замыкая множество, содержащее только одно начальное состояние E, будет получено начальное состояние D. Это основное отличие используемого механизма от конструкции множеств.

(3) F_D – множества состояний, которые содержат, по крайней мере, одно заключительное состояние автомата E. $F_D = \{S | S \subseteq Q_D, S \cap F_E \neq \varnothing\}$.

(4) Функция переходов вычисляется следующим образом:

- Пусть $S = \{p_1, p_2, \ldots, p_k\}$.
- Вычисляем $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \ldots, r_m\}$.
- Получаем $\delta_D(S, a) = \bigcup_{j=1}^{m} ECLOSE(r_j)$. Что и требовалось.

Пример 24.

Попробуем устранить ε-переходы в автомате из примера 18, далее он будет называться E. По нему построим ДКА D, приведенный на СЛАЙДЕ 35.

Т.к. начальное состояние E – это q_0, то начальным состоянием D будет $ECLOSE(q_0)$, т.е. $\{q_0, q_1\}$. Таким образом, сначала нужно найти состояния, в которые переходят q_0 и q_1 по различным входным символам, а у нас это плюс, минус, десятичная точка и цифры от 0 до 9. При решении примера 18 мы видели, что из q_1 по «плюсу» и «минусу» перехода нет (точнее – переход сам в себя), а из q_0 есть переход в q_1. Очевидно, что для вычисления функции $\delta_D(\{q_0, q_1\}, +)$ нужно взять замыкание множества $\{q_1\}$. Но в нашем КА нет ε-переходов, выходящих из q_1, поэтому получаем $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$. Аналогично – для «минуса». Два перехода показаны одной дугой на СЛАЙДЕ 35.

Проделаем те же действия для $\delta_D(\{q_0, q_1\}, .)$. По точке q_0 переходит сам в себя, а q_1 переходит в q_2. Значит, берем замыкание множества $\{q_2\}$, которое является собственным замыканием из-за отсутствия пустых переходов.

Аналогично работаем с цифрами, например, 2. По цифре q_0 переходит сам в себя, а q_1 переходит сразу в q_1 и q_4. Ни у одного из этих состояний нет выходящих ε-переходов, значит, $\delta_D(\{q_0, q_1\}, 2) = \{q_1, q_4\}$. Ясно, что это справедливо и для остальных цифр.

Другие переходы строятся аналогично. Поскольку q_5 – единственное заключительное состояние для E, то заключительными состояниями D являются те его достижимые состояния, которые содержат q_5. В нашем случае это состояния $\{q_3, q_5\}$ и $\{q_2, q_3, q_5\}$, отмеченные двойным кружком.

Теорема 3.3. Язык L допускается некоторым ε-НКА, если он допускается некоторым ДКА (СЛАЙДЫ 36-40).

Доказательство. Достаточность этого утверждения доказать просто. Допустим, $L = L(D)$ для некоторого ДКА D. Преобразуем D в E, добавив переходы $\delta(q, \varepsilon) = \varnothing$ для всех состояний q автомата D. Также преобразуем переходы D к виду НКА-переходов. Т.е., скажем, $\delta_D(q, a) = p$ нужно превратить в множество, содержащее лишь состояние p. Это $\delta_E(q, a) = \{p\}$. Тогда D и E имеют одни и те же не ε-переходы.

Чуть сложнее доказывается необходимость. Применим модифицированную конструкцию подмножеств для построения ДКА, т.е. для D нужно доказать $L(E) = L(D)$, показав, что $\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_0, w)$, где w – произвольная строка из символов алфавита.

Если $|w| = 0$, то $w = \varepsilon$. По определению замыкания $\hat{\delta}_E(q_0, \varepsilon) = ECLOSE(q_0)$. По определению начального состояния ДКА $q_D = ECLOSE(q_0)$. Нам также известно, что для всякого ДКА $\hat{\delta}_D(p, \varepsilon) = p$, каково бы не было состояние p. Это означает, в частности, что $\hat{\delta}_D(q_D, \varepsilon) = ECLOSE(q_0)$. Мы доказали, что $\hat{\delta}_E(q_0, \varepsilon) = \hat{\delta}_D(q_D, \varepsilon)$.

По сложившейся традиции предположим, что $w = xa$, где a – последний символ в строке w, а также то, что для x наше утверждение справедливо. Значит, $\hat{\delta}_E(q_0, x) = \hat{\delta}_D(q_D, x) = \{p_1, p_2, \ldots, p_m\}$.

По определению расширенной функции переходов для ε-НКА вычисляем ее так:

(1) $\bigcup\limits_{i=1}^{k} \delta_E(p_i, a) = \{r_1, r_2, \ldots, r_k\}$.

(2) $\hat{\delta}_E(q_0, w) = \bigcup\limits_{j=1}^{m} ECLOSE(r_j)$

Выше мы рассмотрели построение ДКА D с использованием модифицированной конструкции подмножеств. С учетом этого, мы видим, что $\hat{\delta}_D(\{p_1, p_2, \ldots, p_k\}, a)$ построено с помощью шагов (1) и (2). Следовательно, значение $\hat{\delta}_D(q_D, w) = \hat{\delta}_D(\{p_1, p_2, \ldots, p_k\}, a)$ совпадает с $\hat{\delta}_E(q_0, w)$, т.е. $\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_D, w)$. Как раз то, чего мы и добивались.

Литература к лекции 3

1. Гилл, А. Введение в теорию конечных автоматов / А. Гилл. – М.: Наука, 1966. – 272 с.
2. Кузнецов, А.С. Теория вычислительных процессов [Текст] : учеб. пособие / А. С. Кузнецов, М. А. Русаков, Р. Ю. Царев ; Сиб. федерал. ун-т. - Красноярск: ИПК СФУ, 2008. – 184 с.
3. Короткова, М.А. Математическая теория автоматов. Учебное пособие / М.А. Короткова. – М.: МИФИ, 2008. – 116 с.
4. Молчанов, А. Ю. Системное программное обеспечение. 3-е изд. / А.Ю. Молчанов. –

СПб.: Питер, 2010. – 400 с.
5. Пример реализации конечных автоматов на языке C++ - http://www.devexp.ru/2011/02/konechnye-avtomaty-v-c/
6. Теория автоматов / Э. А. Якубайтис, В. О. Васюкевич, А. Ю. Гобземис, Н. Е. Зазнова, А. А. Курмит, А. А. Лоренц, А. Ф. Петренко, В. П. Чапенко // Теория вероятностей. Математическая статистика. Теоретическая кибернетика. — М.: ВИНИТИ, 1976. — Т. 13. — С. 109–188. — URL http://www.mathnet.ru/php/getFT.phtml?jrnid=intv&paperid=28&what=fullt&option_lang=rus
7. Серебряков В. А., Галочкин М. П., Гончар Д. Р., Фуругян М. Г. Теория и реализация языков программирования — М.: МЗ-Пресс, 2006 г., 2-е изд. - http://trpl7.ru/t-books/TRYAP_BOOK_Details.htm
8. Недетерминированные конечные автоматы - http://www.rsdn.ru/article/alg/nka.xml
9. Finite State Machine Generator - http://sourceforge.net/projects/genfsm/
10. Введение в схемы, автоматы и алгоритмы - http://www.intuit.ru/studies/courses/1030/205/info