Домашняя работа № 5

Автор: Минеева Екатерина

Задача A. Минимизация автомата

Главные идеи:

0. Выкидываем все состояния, которые недостижимы из начального – они никак не влияют на работу автомата.

1. Пытаемся разбить состояния на классы эквивалентности: состояния из одного класса мы можем объединить в одну и получить эквивалентный автомат, а для состояний из разных классов существуют две строки $u \in L$, $v \notin L$ (L – язык, распозноваемый данным автоматом A.), такие, что при объединении этих двух состояний в одно, результат работы автомата A на строках u и v будет одинаковый.

Подробнее про пункт 1:

Изначально разделяем все состояния на два класса эквивалентности: финальные и нефинальные. Очевидно, что финальное и нефинальное состояние не могут оказаться в одном классе эквивалентности.

Далее пытаемся понять, какие неэквивалентные состояния мы объединили в одно. Для каждой пары неэквивалентных состояний мы проверяем, есть ли такие два состояния, что на данный момент они находятся в одном классе, но при этом по одному и тому же символу из них осуществляется переход в наши состояния, которые не являются эквивалентными.

Детали реализации:

Заполним двумерную таблицу $notEquivalent$, где $notEquivalent[i][j] = true$, если состояния i и j находятся в разных классах эквивалентности, и $false$ иначе.

Сначала помечаем, как неэквивалентные, пары состояний финальное-нефинальное. После того, как мы отметили очередную пару состояний, как неэквивалентные, какие-то другие пары состояний тоже могут стать неэквивалентными. Поэтому используем очередь, в которую будем помещать каждую пару ранее эквивалентных состояний, которые мы отметили, как неэквивалентные. После чего будем извлекать из очереди пары состоний q_1, q_2, и проверять все пары состояний $(q'_1,\ q'_2)$ такие, что по какому-то символу алфавита, мы могли попасть из q_1 в q'_1, и q_2 в q'_2.

Если в какой-то момент очередь оказалась пуста, значит таблица оказалась построена. А по таблице $notEquivalent$ уже не составляет труда разбить множество исходных состояний на классы эквивалентности. Минимальность автомата, в котором состояния отвечают классам эквивалентности, на которые распадаются состояния исходного автомата, следует из самого определения классов эквивалентности для состояний.

Оценка сложности:

Пусть n – число состояний исходного автомата, l – размер алфавита.
За $\underline{\text{O}}(n^2)$, помечаем, как неэквивалентные, пары финальное-нефинальное состояние и кладем в очередь.

Пока очередь не пуста, достаем оттуда пару состояний q_1, q_2 и проверяем пары состояний $(q'_1,\ q'_2)$ такие, что по какому-то $s \in \Sigma$, можно попасть из q_1 в q'_1, и q_2 в q'_2. Заметим, что каждая пара состояний q_1, q_2 может попасть в очередь не более одного раза. Различных символов – l. Количество различных символов q'_1, q'_2– размеры множеств $\delta^{-1}(q_1, s)$ и $\delta^{-1}(q_2, s)$ соответственно (δ – функция перехода):

$$\underline{\text{O}}\left(\sum_{i,j=1}^{n}\sum_{s\in\Sigma}|\delta^{-1}(q_i,s)|\cdot|\delta^{-1}(q_j,s)|\right) = \underline{\text{O}}\left(\sum_{s\in\Sigma}\sum_{i,j=1}^{n}|\delta^{-1}(q_i,s)|\cdot|\delta^{-1}(q_j,s)|\right) =$$

$$= \underline{\text{O}}\left(l\cdot\sum_{i,j=1}^{n}|\delta^{-1}(q_i,s)|\cdot|\delta^{-1}(q_j,s)|\right) \leq \underline{\text{O}}\left(l\cdot\sum_{i=1}^{n}|\delta^{-1}(q_i,s)|\cdot\sum_{j=1}^{n}|\delta^{-1}(q_j,s)|\right) = \underline{\text{O}}\left(n^2 l\right)$$

Вычисление количества классов эквивалентности по $notEquivalent$ – $\underline{\text{O}}(n^2)$. Итого сложность $\underline{\text{O}}\left(n^2 l\right)$