Оптимизация автоматов

При использовании общего алгоритма детерминизации автоматов могут возникнуть вспомогательные состояния. Такие состояния можно убрать, сохранив выводимый язык.

Обратим внимание особый тип вершин в визуальном представлении графа – вершины, из которых **выходит ровно одна λ-дуга**. Если это не финальная вершина, то нам нет смысла на ней останавливаться и можем всегда пройти на последующие вершины. А значит, она не влияет на текущее состояние.

Если в вершину **входит ровно одна λ-дуга**, это значит, мы всегда можем пройти в будущем через неё из предыдущей вершины. Поэтому она не влияет на текущее состояние.

Назовём такие вершины *проходящими* – вершины, в которые входит или из которых выходит ровно одна λ-дуга.

Раз они не влияют на состояния конечного автомата, сделаем преобразование, убирающее их из конечных расчётов.

Введём *отношение эквивалентости* между состояниями. Два состояния назовём эквивалентными, если они отличаются лишь проходящими вершинами. Такие состояния будут группироваться в классы эквивалентности.

*Оптимизированными состояниями* назовём те, которые из состоят из классов эквивалентности.

Пример:

|  |  |  |
| --- | --- | --- |
|  | 0 | 9 |
| q1 = { 1 } |  | q2 |
| [q2] = { 2, [3] } | q3 | q3 |
| [q3] = { [3] } | q3 | q3 |

Пусть у нас есть три состояния qi, состоящие из вершин λ-диаграммы. Известно, что вершина 2 – проходящая.

Тогда [q2] ~ [q3].

Получаем два класса эквивалентности: r1 = { q1 }, [r2] = { [q2], [q3] }.

Объединим строки, принадлеждащие к одному и тому же классу эквивалентности.

|  |  |  |
| --- | --- | --- |
|  | 0 | 9 |
| r1 = { q1 } |  | q2 |
| r2 = { [q2], [q3] } | q3, q3 | q3, q3 |

И заменим значения в ячейках на эквивалентные классы:

q1 -> r1, q2 -> r2, q3 -> r2

|  |  |  |
| --- | --- | --- |
|  | 0 | 9 |
| r1 = { q1 } |  | r2 |
| r2 = { [q2], [q3] } | r2 | r2 |

Получили таблицу *оптимизированного* ДКА.