**Методический материал для лабораторной работы №3 . Алгоритм преобразования НДКА в ДКА.**

Например, для грамматики **G = 〈{a, b, ⊥}, {S, A, B}, P, S 〉**, где  
P:  
S → A⊥  
A → a | Bb  
B → b | Bb  
разбор будет недетерминированным (т.к. у нетерминалов A и B есть одинаковые правые части Bb). Такой грамматике будет соответствовать недетерминированный конечный автомат.  
**Определение:** **недетерминированный конечный автомат** (НКА) — это пятерка 〈 K, T,δ, H, S 〉, где  
K — конечное множество состояний;  
T — конечное множество допустимых входных символов;  
δ — отображение множества K × T в множество подмножеств K;  
H ⊂ K — конечное множество начальных состояний;  
S ⊂ K — конечное множество заключительных состояний.  
δ(A, t) = {B1, B2,..., Bn} означает, что из состояния A по входному символу t можно осуществить переход в любое из состояний Bi, i = 1, 2, ..., n. Также как и ДКА, любой НКА можно  
представить в виде таблицы (в одной ячейке такой таблицы можно указывать сразу несколько состояний, в которые возможен переход из заданного состояния по текущему символу)  
или в виде диаграммы состояний (ДС). В ДС каждому состоянию из множества K соответствует вершина; из вершины A в вершину B ведет дуга, помеченная символом t, если  
B ∈ δ (A, t). (В НКА из одной вершины могут исходить несколько дуг с одинаковой пометкой). Успешный путь — это путь из начальной вершины в заключительную; пометка пути  
— это последовательность пометок его дуг.

**Язык, допускаемый НКА, — это множество пометок всех успешных путей.**Если начальное состояние автомата (НКА или ДКА) одновременно является и заключительным, то автомат допускает пустую цепочку ε.  
**Замечание.**Автомат, построенный по регулярной грамматике без пустых правых частей, не допускает ε.  
Для построения разбора по регулярной грамматике в недетерминированном случае  
можно предложить алгоритм, который будет перебирать все возможные варианты сверток  
(переходов) один за другим; если цепочка принадлежит языку, то будет найден успешный  
путь; если каждый из просмотренных вариантов завершится неудачей, то цепочка языку не  
принадлежит. **Однако такой алгоритм практически неприемлем, поскольку при переборе вариантов мы, скорее всего, снова окажемся перед проблемой выбора и, следовательно, будем иметь «дерево отложенных вариантов» и экспоненциальный рост сложности разбора.**Один из наиболее важных результатов теории конечных автоматов состоит в том,  
что класс языков, определяемых недетерминированными конечными автоматами, совпадает  
с классом языков, определяемых детерминированными конечными автоматами.

**Алгоритм построения детерминированного КА по НКА**

**Вход:** M = (K, VT, F, H, S) - недетерминированный конечный автомат.  
**Выход:** M’ = (K’, VT, F’, H’, S’) - детерминированный конечный автомат,  
допускающий тот же язык, что и автомат М.  
**Метод:**1. Множество состояний К’ состоит из всех подмножеств множества К.  
Каждое состояние из К’ будем обозначать [A1A2 ...An ], где Ai ∈ K.  
2. Отображение F’ определим как F’ ([A1A2 ...An ], t) = [B1B2 ...Bm], где для  
каждого 1 <= j <= m F(Ai , t) = Bj для каких-либо 1 <= i <= n.  
3. Пусть H = {H1, H2 , ..., Hk }, тогда H’ = [H1, H2 , ..., Hk ].  
4. Пусть S = {S1, S 2 , ..., Sp }, тогда S’ - все состояния из K’, имеющие вид  
[...Si ...], .S i ∈ S для какого-либо 1 <= i <= p.  
**Замечание:** в множестве K’ могут оказаться состояния, которые недостижимы  
из начального состояния, их можно исключить.

**Проиллюстрируем работу алгоритма на примерах.**

**Пример 1.** Задан НКА M = 〈{ H, A, B, S }, {0, 1}, δ, {H }, {S } 〉, где  
δ(H, 1) = {B}  
δ(A, 1) = {B, S}  
δ(B, 0) = { A }  
L(M) = { 1(01)n | n ≥ 1 }. Диаграмма для M изображена на рис.1

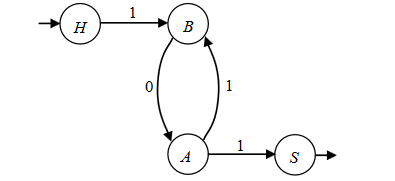


Рис 1 . Диаграмма состояний для автомата М

**Грамматика, соответствующая M:**S → A1  
A → B0  
B → A1 | 1

Построим ДКА по НКА, пользуясь предложенным алгоритмом. Начальным состоянием будет H.  
δ1(Н, 1) = B  
δ1(B, 0) = A  
δ1(A, 1) = BS  
δ1(**BS**, 0) = A  
Заключительным состоянием построенного ДКА является состояние **BS**, так как содержит заключительное состояние из множества заключительных состояний НКА ( S – заключительное состояние в НКА).

Таким образом, M1 = 〈 {H, B, A, BS }, {0, 1}, δ1, H, BS 〉. Для удобства переименуем состояния в M1 : BS обозначается теперь как S1, а в однобуквенных именах состояний вместо подчеркивания используется индекс 1. Тогда M1 = 〈 {H1, B1, A1, S1}, {0, 1}, { δ1(Н1, 1) = B1; δ1(B1, 0) = A1; δ1(A1, 1) = S1; δ1(S1, 0) = A1}, H1, S1〉

**Грамматика, соответствующая M1:**S1 → A11  
A1 → S10 | B10  
B1 → 1  
Построим диаграмму состояний М1.

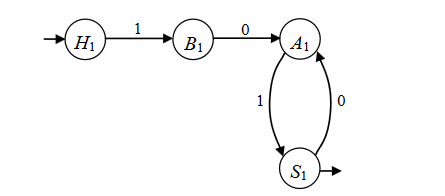
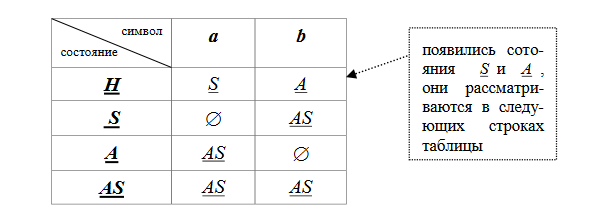


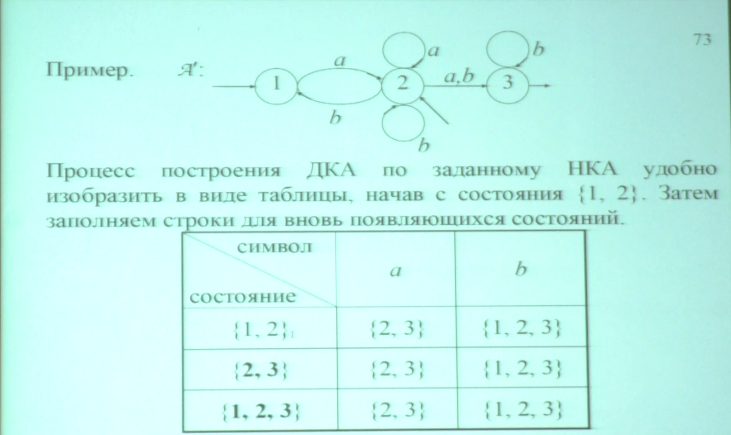
Рис 2 . Диаграмма состояний для автомата М1

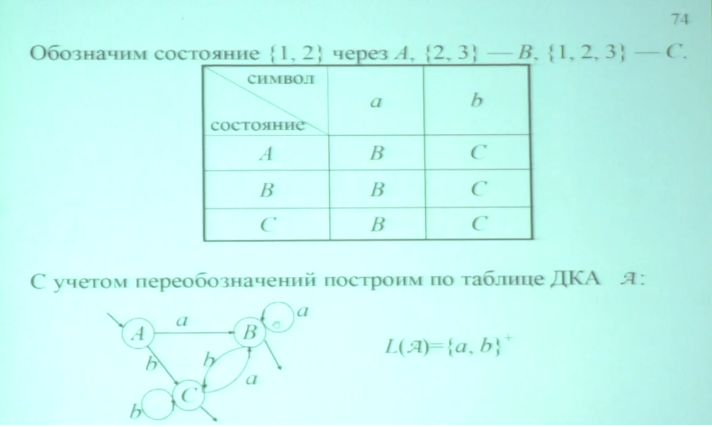
**Пример 2.** Дана регулярная грамматика G = 〈 T, N, P, S 〉  
P:  
S → Sb | Aa | a  
A → Aa | Sb | b,  
которой соответствует НКА.

*С помощью преобразования НКА в ДКА построить грамматику, по которой возможен детерминированный разбор.*  
**Построим функцию переходов НКА:**  
δ(H, a) = {S}  
δ(H, b) = {A}  
δ(A, a) = {A, S}  
δ(S, b) = {A, S}  
**Процесс построения функции переходов ДКА удобно изобразить в виде таблицы,  
начав ее заполнение с начального состояния H, добавляя строки для вновь появляющихся  
состояний:**

Переобозначим состояния: A ⇒ A1, H ⇒ H1, AS ⇒ Y, S ⇒ X. Два заключительных  
состояния X и Y сведем в одно заключительное состояние S1, используя маркер конца цепочки ⊥. После такой модификации получаем ДКА с единственным заключительным состоянием S1 и функцией переходов:  
δ1(H1, a) = X  
δ1(H1, b) = A1  
δ1(X, b) = Y  
δ1(X, ⊥) = S1  
δ1(A1, a) = Y  
δ1(Y, a) = Y  
δ1(Y, b) = Y  
δ1(Y,⊥) = S1  
По ДКА строим грамматику **G1**, позволяющую воспользоваться алгоритмом детерминированного разбора:  
**G1:**S1 → X⊥ | Y⊥  
Y → Ya | Yb | A1a | Xb  
X → a  
A1 → b

**Пример 3.**



****