

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних інформаційних систем
Алгебро-автоматичні методи проектування програмного забезпечення

Лабораторна робота 1
“Алгоритм детермінації Х-автомата”

Виконали студенти 1-го курсу

Групи ПЗС-1

Богатько Олександр Геннадійович

Юзина Сергій Сергійович

Полосенко Павло Олегович

Мета: Перетворити недетермінований скінченний автомат (НСА) на детермінований скінченний автомат (ДСА).

Псевдокод

НДСА-ДСА(\mathcal{A})

Вхід: Недетермінований СА $\mathcal{A} = (A, X, Q, a_0, F)$.

Вихід: Детермінований СА $\mathcal{B} = (B, X, f, b_0, F')$.

Метод:

1. $B := \emptyset$; $F' := \emptyset$; $b_0 := a_0$;
2. $C := \{b_0\}$;
3. while $C \neq \emptyset$ do
4. pick C' from C ;
5. $B := B \cup C'$;
6. if $C' \cap F \neq \emptyset$ then $F' := F' \cup C'$;
7. for all $x \in X$ do
8. $C'' := \bigcup_{a \in C'} Q(a, x)$;
9. if $C'' \notin B$ then $C := C \cup C''$;
10. $f := f \cup (C', x, C'')$;
11. od
12. od

Код програми:

<https://github.com/sbohatko/MastersLabs/blob/main/algebra-automatic-software-design-methods/lab1.py>

Результат виконання програми:

```
alexbogatk@MacBook-Pro-Alex ~/D/M/algebra-automatic-software-design-methods (main)> python3 lab1.py
NFA: {'A': ['1', '2', '3', '4'], 'X': ['a', 'b', 'c'], 'f': {'(1', 'b)': ['2', '3'], ('2', 'a)': ['2'], ('3', 'c)': ['4'], ('4', 'a)': ['4'], ('1', 'a)': ['2'], ('1', 'c)': ['4'], ('2', 'b)': ['2', '3']}, 'a0': '1', 'F': ['1', '3', '4'], 'epsilon': 'e'}

DFA: {'A': [('1',), ('2',), ('2', '3'), ('4',)], 'X': ['a', 'b', 'c'], 'f': {(('1',), 'a'): ('2',), (('1',), 'b'): ('2', '3'), (('1',), 'c'): ('4',), (('2',), 'a'): ('2',), (('2',), 'b'): ('2', '3'), (('2', '3'), 'a'): ('2',), (('2', '3'), 'b'): ('2', '3'), (('2', '3'), 'c'): ('4',), (('4',), 'a'): ('4',)}, 'a0': ('1',), 'F': [('1',), ('2', '3'), ('4',)]}

DFA States:
('1',)
('2',)
('2', '3')
('4',)

Alphabet (X):
a
b
c

Transitions (f):
Transition from ('1',) with input a goes to ('2',)
Transition from ('1',) with input b goes to ('2', '3')
Transition from ('1',) with input c goes to ('4',)
Transition from ('2',) with input a goes to ('2',)
Transition from ('2',) with input b goes to ('2', '3')
Transition from ('2', '3') with input a goes to ('2',)
Transition from ('2', '3') with input b goes to ('2', '3')
Transition from ('2', '3') with input c goes to ('4',)
Transition from ('4',) with input a goes to ('4',)

Start State (a0):
('1',)

Final States (F):
('1',)
('2', '3')
('4',)
```

Пояснення

Цей код реалізує алгоритм конструкції підмножин для перетворення недетермінованого скінченного автомату (НСА) у детермінований скінчений автомат (ДСА).

Давайте розглянемо алгоритм по крокам:

Функція `epsilon_closure(states, transitions, epsilon='e')`:

Ця функція обчислює епсілон-замикання для заданої множини станів в НСА. Епсілон-замикання для стану s - це множина станів, які можна досягти з s за допомогою переходів по епсілону.

1. $e_closure$ ініціалізується як множина вхідних станів.
2. Створюється стек $stack$, в який додаються вхідні стани.
3. Поки стек не порожній:
 - Вилучається стан зі стеку ($state$).
 - Формується ключ ($state, epsilon$).
 - Якщо такий ключ присутній у переходах ($transitions$), то для кожного наступного стану:
 - Якщо цей стан ще не входить у $e_closure$, він додається до нього і додається до стеку.

Функція $nfa_to_dfa(nfa)$:

Ця функція використовує алгоритм побудови підмножин для конвертації НСА у ДСА.

1. Вона отримує необхідну інформацію з НСА: стани, алфавіт, переходи, стартовий стан та кінцеві стани.
2. Ініціалізує списки і словники для ведення обліку станів та переходів ДСА.
3. Обчислює епсілон-замикання початкового стану і додає його як початковий стан ДСА.
4. У циклі для кожного необробленого стану:
 - Для кожного символу алфавіту:
 - Обчислюється множина наступних станів, до яких можна дістатися з поточного стану за допомогою цього символу.
 - Обчислюється епсілон-замикання цієї множини.
 - Якщо така множина ще не присутня в станах ДСА, вона додається, а також вноситься в список необроблених станів.
 - Записується перехід в словник $dfa_transitions$.
5. Визначаються кінцеві стани ДСА шляхом перевірки, чи присутні якісь з кінцевих станів НСА в станах ДСА.

6. Повертається результуючий ДСА у вигляді словника, що включає стани, алфавіт, переходи, початковий стан та кінцеві стани.

Складність алгоритму

Алгоритм, який використовується в цьому коді, базується на алгоритмі побудови підмножини для перетворення недетермінованого кінцевого автомата (NFA) у детермінований кінцевий автомат (DFA).

Часова складність алгоритму побудови підмножини становить $O(2^n * m)$, де:

- **n** – кількість станів у результуючому DFA.
- **m** – кількість символів в алфавіті.

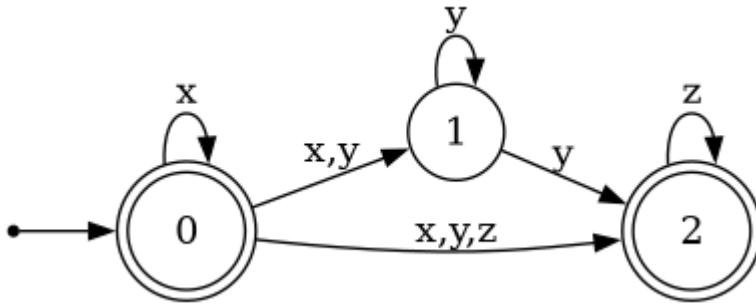
Ось коротке пояснення:

- **Експоненціальне зростання:** у гіршому випадку кількість станів у отриманому DFA може бути експоненціальною щодо кількості станів у вихідному NFA. Це тому, що кожна підмножина станів у NFA відповідає стану в DFA.
- **Вкладений цикл:** алгоритм включає вкладені цикли. Для кожного стану в DFA він повторює кожен символ в алфавіті та обчислює набір станів, яких можна досягти. Це сприяє ускладненню часу.
- **Епсилонне закриття:** функція епсилонного замикання також передбачає ітерацію станів і переходів, що може зайняти багато часу залежно від структури NFA.
- **Операції з множинами:** операції з множинами (додавання, об'єднання тощо) беруть участь у створенні станів і переходів DFA.

Загалом, складність алгоритму в першу чергу визначається кількістю станів у результуючому DFA та розміром алфавіту. Це може бути досить ефективним для невеликих NFA, але може стати обчислювально дорогим для більших і складніших NFA.

Приклад 1:

Недетермінований автомат



Перехід від (0) з введенням x переходить до ('0', '1', '2')

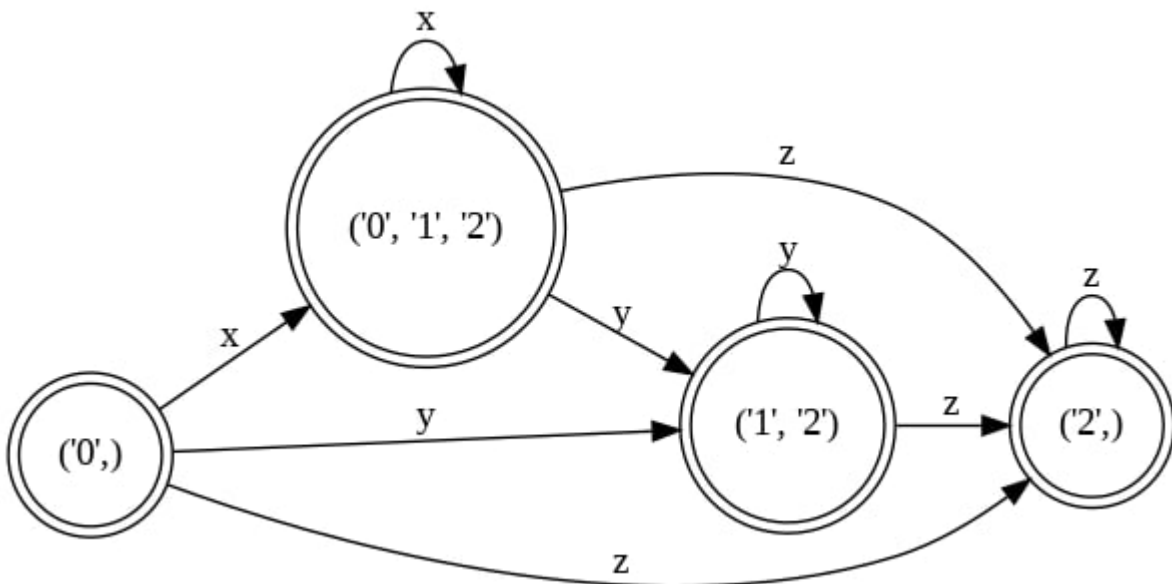
Перехід від (0) з введенням y переходить до ('1', '2')

Перехід від (0) з введенням z переходить до ('2')

Перехід від (1) з введенням y переходить до ('1', '2')

Перехід від (2) з введенням z переходить до ('2')

Після застосування алгоритму отримуємо такий детермінований автомат



Перехід від ('0',) з введенням x переходить до ('0', '1', '2')

Перехід від ('0',) з введенням y переходить до ('1', '2')

Перехід від ('0',) з введенням z переходить до ('2',)

Перехід від ('0', '1', '2') з введенням x переходить до ('0', '1', '2')

Перехід від ('0', '1', '2') з введенням y переходить до ('1', '2')

Перехід від ('0', '1', '2') з введенням z переходить до ('2',)

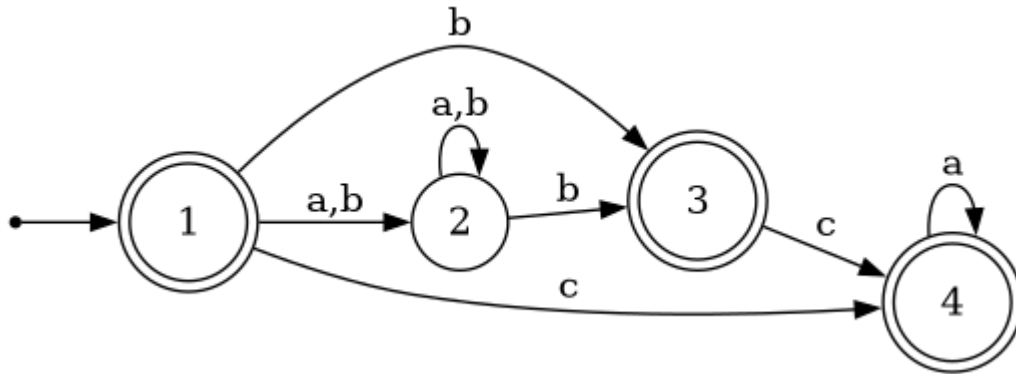
Перехід від ('1', '2') з введенням y переходить до ('1', '2')

Перехід від ('1', '2') з введенням z переходить до ('2',)

Перехід від ('2',) з введенням z переходить до ('2',)

Приклад 2:

Недетермінований автомат



Перехід від (1) з введенням a переходить до ('2')

Перехід від (1) з введенням b переходить до ('2', '3')

Перехід від (1) з введенням c переходить до ('4')

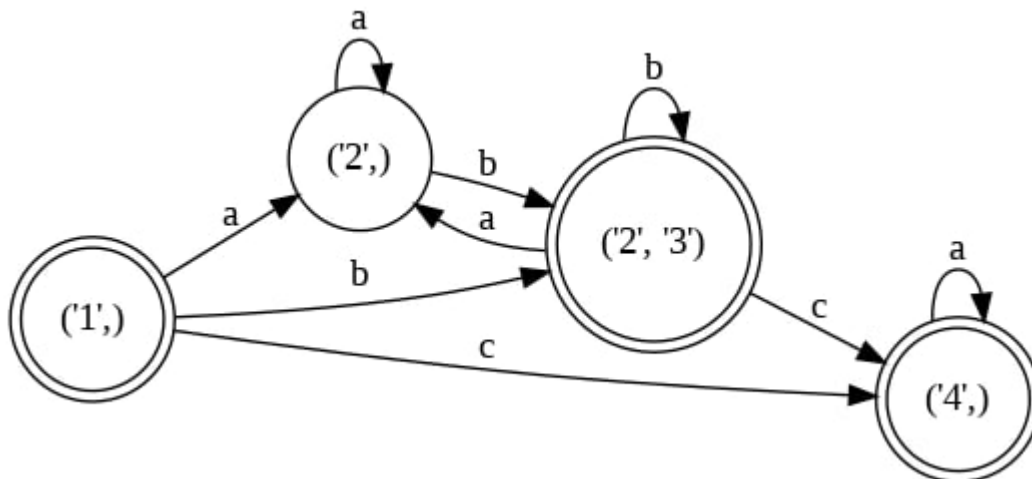
Перехід від (2) з введенням a переходить до ('2')

Перехід від (2) з введенням b переходить до ('2', '3')

Перехід від (3) з введенням c переходить до ('4')

Перехід від (4) з введенням a переходить до ('4')

Після застосування алгоритму отримуємо такий детермінований автомат



Перехід від ('1,') з введенням a переходить до ('2,')

Перехід від ('1,') з введенням b переходить до ('2, '3')

Перехід від ('1,') з введенням c переходить до ('4,')

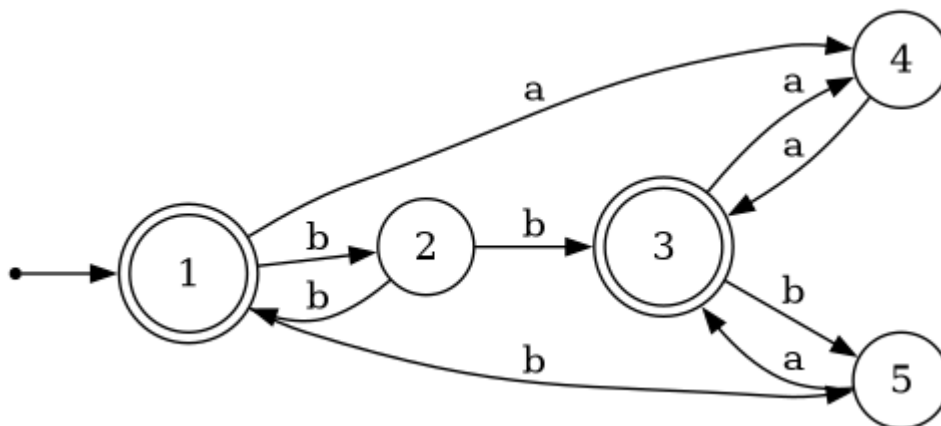
Перехід від ('2,') з введенням a переходить до ('2,')

Перехід від ('2,') з введенням b переходить до ('2, '3')

Перехід від ('2', '3') з введенням а переходить до ('2',)
 Перехід від ('2', '3') з введенням b переходить до ('2', '3')
 Перехід від ('2', '3') з введенням с переходить до ('4',)
 Перехід від ('4',) з введенням а переходить до ('4',)

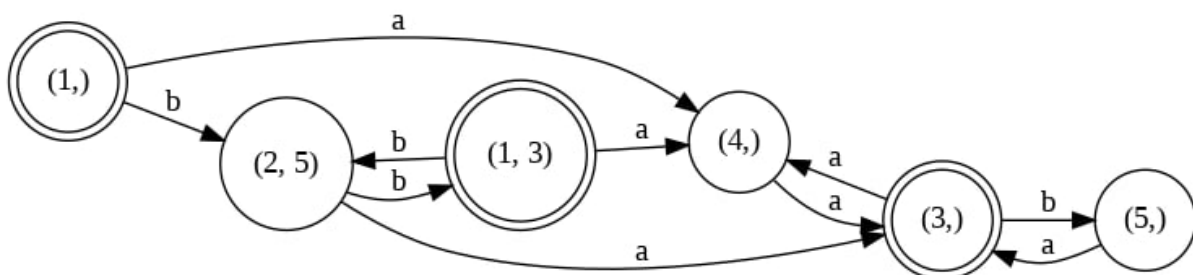
Приклад 3:

Недетермінований автомат



Перехід від (1) з введенням а переходить до ('4')
 Перехід від (1) з введенням b переходить до ('2', '5')
 Перехід від (2) з введенням b переходить до ('1', '3')
 Перехід від (3) з введенням а переходить до ('4')
 Перехід від (3) з введенням b переходить до ('5')
 Перехід від (4) з введенням а переходить до ('3')
 Перехід від (5) з введенням а переходить до ('3')

Після застосування алгоритму отримуємо такий детермінований автомат

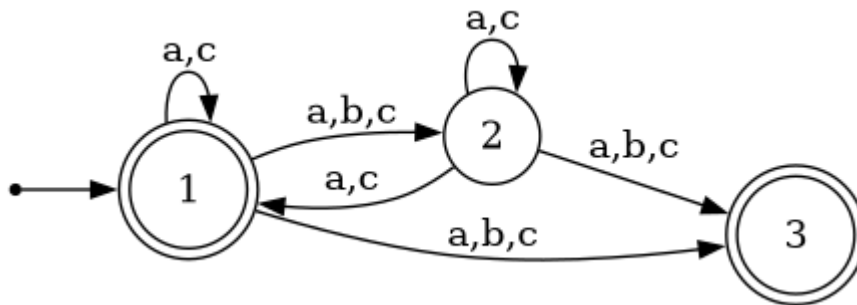


Перехід від (1,) з введенням а переходить до (4,)
 Перехід від (1,) з входом b переходить до (2, 5)
 Перехід від (4,) з введенням а переходить до (3,)
 Перехід від (2, 5) з введенням а переходить до (3,)
 Перехід від (2, 5) з входом b переходить до (1, 3)
 Перехід від (3,) з введенням а переходить до (4,)

Перехід від (3,) з входом b переходить до (5,)
 Перехід від (1, 3) з введенням a переходить до (4,)
 Перехід від (1, 3) з входом b переходить до (2, 5)
 Перехід від (5,) з введенням a переходить до (3,)

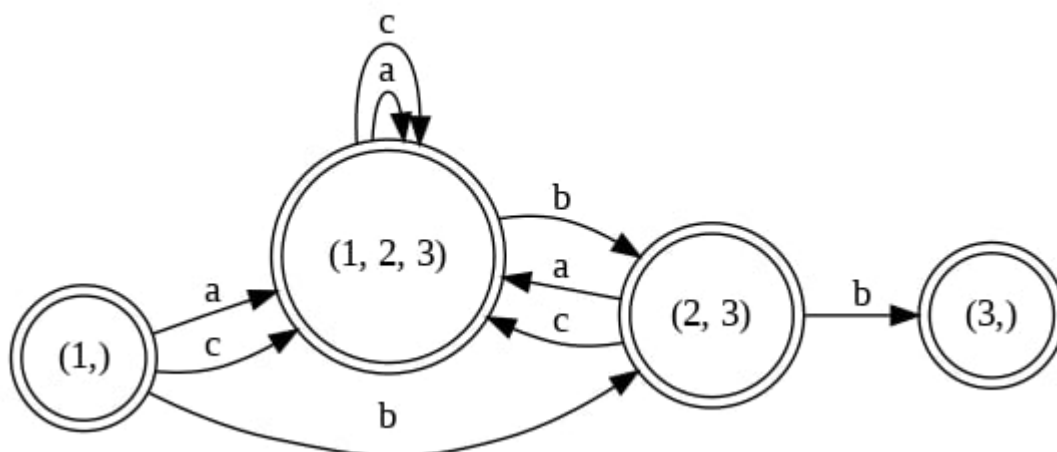
Приклад 4:

Недетермінований автомат



Перехід від (1) з введенням a переходить до ('1', '2', '3')
 Перехід від (1) з введенням b переходить до ('2', '3')
 Перехід від (1) з введенням c переходить до ('1', '2', '3')
 Перехід від (2) з введенням a переходить до ('1', '2', '3')
 Перехід від (2) з введенням b переходить до (3)
 Перехід від (2) з введенням c переходить до ('1', '2', '3')

Після застосування алгоритму отримуємо такий детермінований автомат

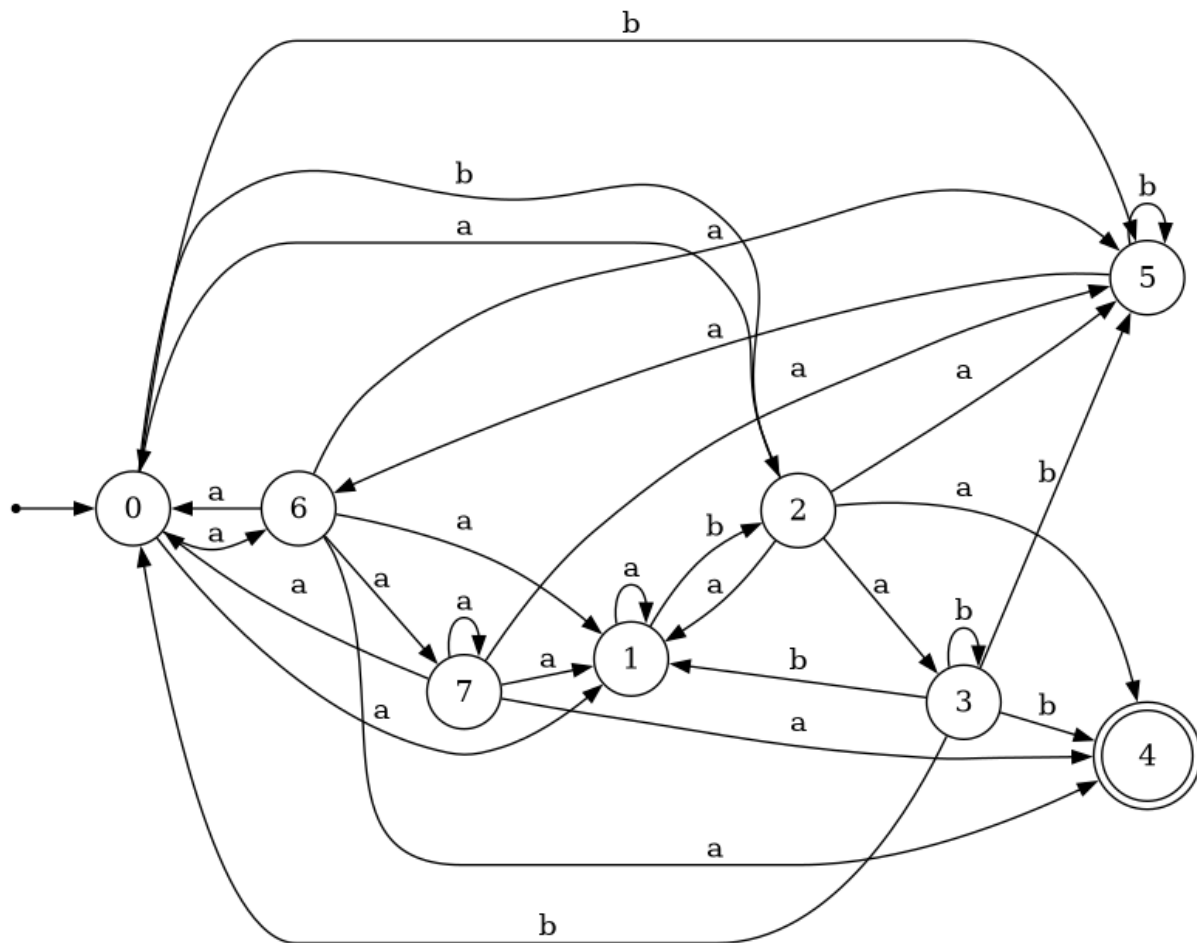


Перехід від (1,) з введенням a переходить до (1, 2, 3)
 Перехід від (1,) з входом b переходить до (2, 3)
 Перехід від (1,) з введенням c переходить до (1, 2, 3)
 Перехід від (1, 2, 3) з введенням a переходить до (1, 2, 3)

Перехід від (1, 2, 3) з введенням b переходить до (2, 3)
 Перехід від (1, 2, 3) з введенням c переходить до (1, 2, 3)
 Перехід від (2, 3) з введенням a переходить до (1, 2, 3)
 Перехід від (2, 3) з входом b переходить до (3,)
 Перехід від (2, 3) з введенням c переходить до (1, 2, 3)

Приклад 5:

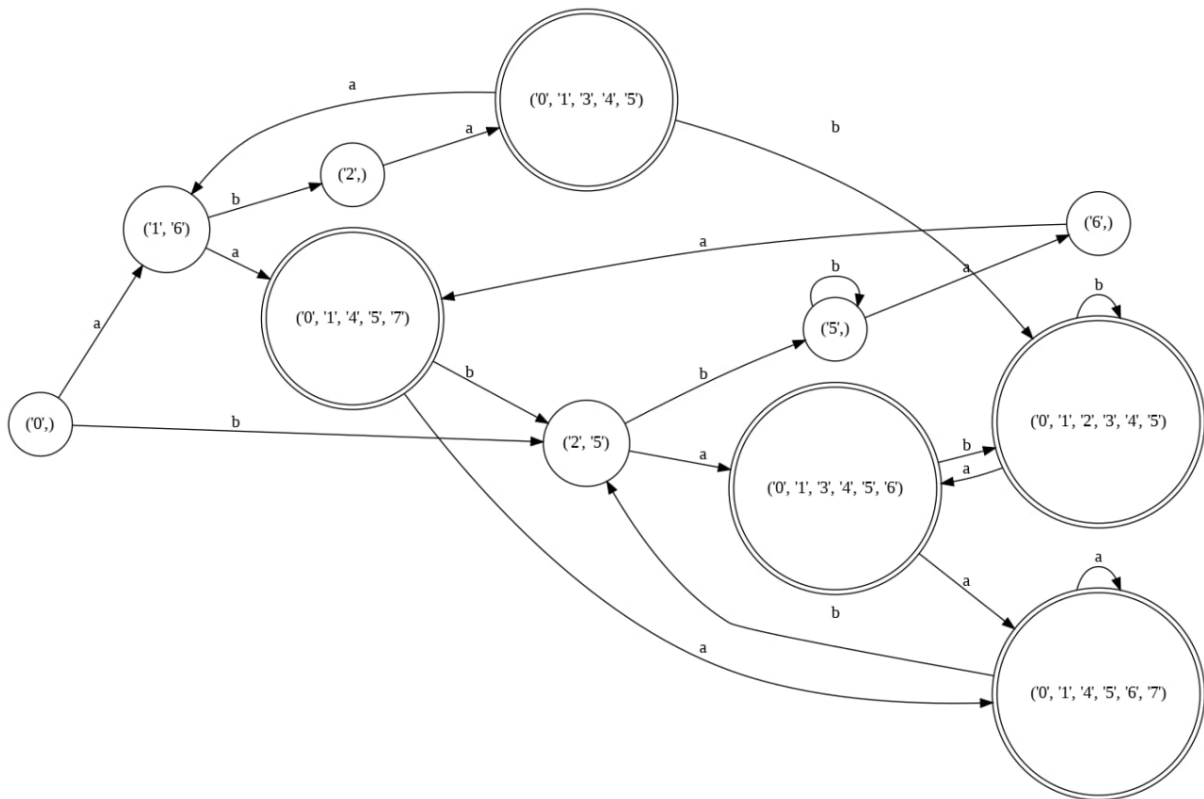
Недетермінований автомат



Перехід від (0) з введенням a переходить до ('1', '6')
 Перехід від (0) з введенням b переходить до ('2', '5')
 Перехід від (1) з введенням a переходить до ('1')
 Перехід від (1) з введенням b переходить до ('2')
 Перехід від (2) з введенням a переходить до ('0', '1', '3', '4', '5')
 Перехід від (3) з введенням b переходить до ('0', '1', '3', '4', '5')
 Перехід від (4) з введенням a переходить до ('0', '1', '4', '5', '7')
 Перехід від (5) з введенням a переходить до ('6')
 Перехід від (5) з введенням b переходить до ('5')
 Перехід від (6) з введенням a переходить до ('0', '1', '4', '5', '7')

Перехід від (7) з введенням а переходить до ('0', '1', '4', '5', '7')

Після застосування алгоритму отримуємо такий детермінований автомат



Перехід від ('0',) з введенням а переходить до ('1', '6')

Перехід від ('0',) з введенням б переходить до ('2', '5')

Перехід від ('1', '6') з введенням а переходить до ('0', '1', '4', '5', '7')

Перехід від ('1', '6') з введенням б переходить до ('2',)

Перехід від ('2', '5') з введенням а переходить до ('0', '1', '3', '4', '5', '6')

Перехід від ('2', '5') з введенням б переходить до ('5',)

Перехід від ('0', '1', '4', '5', '7') з введенням а переходить до ('0', '1', '4', '5', '6', '7')

Перехід від ('0', '1', '4', '5', '7') з введенням б переходить до ('2', '5')

Перехід від ('2',) з введенням а переходить до ('0', '1', '3', '4', '5')

Перехід від ('0', '1', '3', '4', '5', '6') з введенням а переходить до ('0', '1', '4', '5', '6', '7')

Перехід від ('0', '1', '3', '4', '5', '6') з введенням б переходить до ('0', '1', '2', '3', '4', '5')

Перехід від ('5',) з введенням а переходить до ('6',)

Перехід від ('5',) з введенням б переходить до ('5',)

Перехід від ('0', '1', '4', '5', '6', '7') з введенням а переходить до ('0', '1', '4', '5', '6', '7')

Перехід від ('0', '1', '4', '5', '6', '7') з введенням b переходить до ('2', '5')

Перехід від ('0', '1', '3', '4', '5') з введенням a переходить до ('1', '6')

Перехід від ('0', '1', '3', '4', '5') з введенням b переходить до ('0', '1', '2', '3', '4', '5')

Перехід від ('0', '1', '2', '3', '4', '5') з введенням a переходить до ('0', '1', '3', '4', '5', '6')

Перехід від ('0', '1', '2', '3', '4', '5') з введенням b переходить до ('0', '1', '2', '3', '4', '5')

Перехід від ('6',) з введенням a переходить до ('0', '1', '4', '5', '7')

Висновок: Алгоритм надає нам засіб для перетворення НДСА на ДСА, враховуючи всі можливі стани. З цієї роботи ми навчилися основам роботи з НДСА і ДСА та перетворення першого в другий.