جَامعة
المَنارة
MANARA UNIVERSITY

Compilers Techniques
Lecture3
المحاضرة الثالثة

Lexical Analyses (Regular expressions and NFA)

المحلل المعجمي (التعابير المنتظمة وآليتها غير المحددة)

Reference: Aho2, Sections 2.2, 3.1-3.4. Aho1, pp. 25-29; 84-87; 92-105. Hunter, Chapter 2 (too detailed); Sec. 3.1 -3.3 (too condensed). Grune 1.9; 2.1-2.5. Cooper, Sections 2.1-2.3.

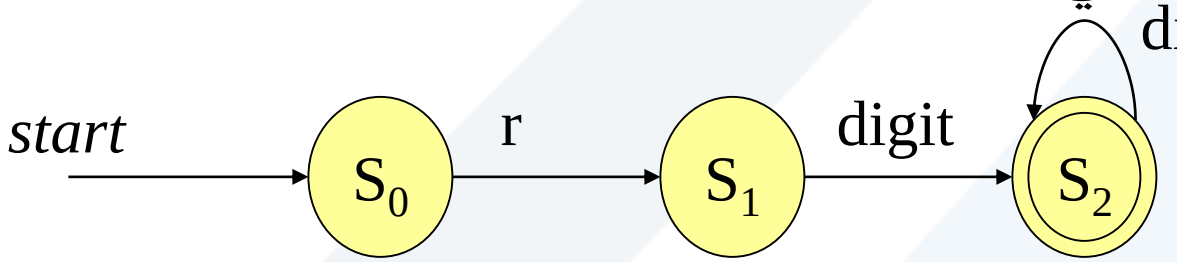
السنة الرابعة –المستوى السابع- الهندسة المعلوماتية

التحليل المعجمي Lexical Analysis

لنأخذ مسألة التعرف على أسماء المسجلات ابتداءً من المسجل r والتي تحتاج إلى رقم عشري وحيد على الأقل.

*Register → r (0|1|2|…|9) (0|1|2|…|9)** (or, *Register → r Digit Digit**)

ويكون لدينا آلية التعبير النظامي وفقاً لمخطط الانتقال هو:



يمكننا التعبير عن الفعاليات التي تحدث في الماسح (المحلل المعجمي) كما يلي:

تمثل الدائرة الحالة.(S0: start state; S2: final state (double circle

يمثل الحرف الانتقال transition. ويمثل العلام label سبب الانتقال.

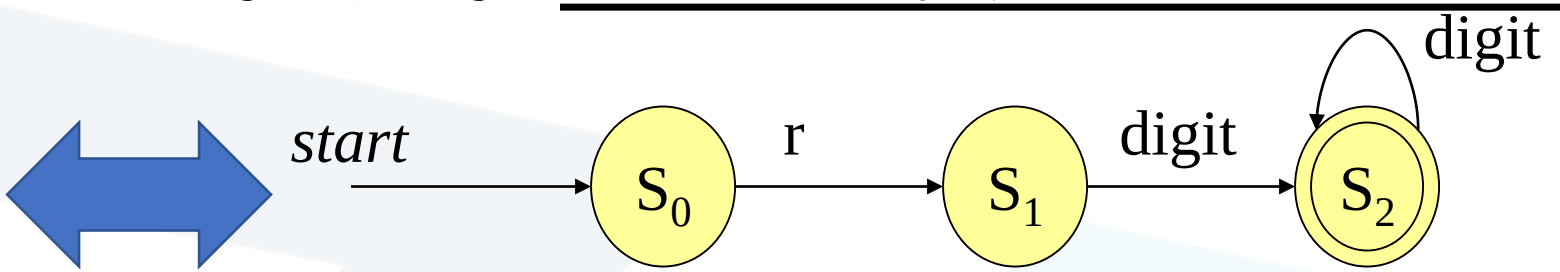
تعتبر السلسلة صحيحة إذا انتهت بعد الانتقالات بحالة نهائيةfinal state. مثل: r345, r0, r29, ,

تعتبر سلسلة الدخل خاطئة إذا لم تنته بعد الانتقالات بحالة نهائية مثلا: a, r, rab

التحليل المعجمي Lexical Analysis

جدول الانتقالات: transition table هو تنفيذ مؤتمت لمخطط الانتقالات.

state	'r'	digit
0	1	-
1	-	2
2(final)	-	2



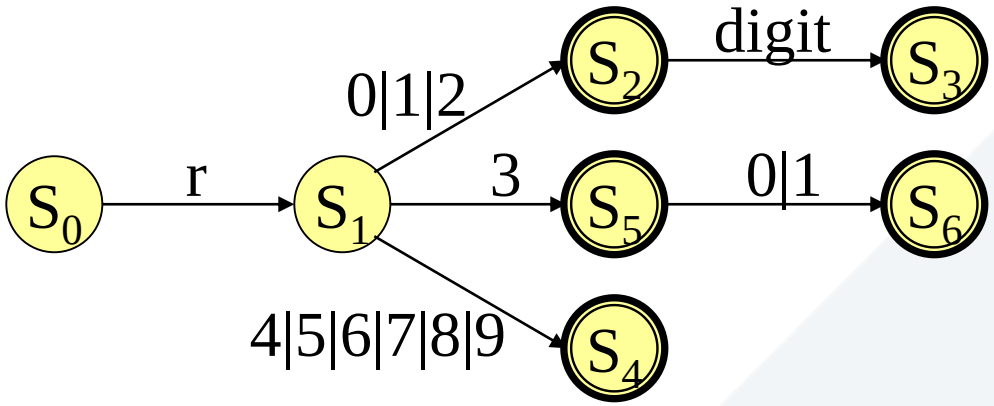
- إذا كان لدينا جدول الانتقال والحالة (أو الحالات) النهائية فإنه بإمكاننا أن نبني بشكل مباشر متعرف recogniser يكتشف إمكانية القبول acceptance:

```
char=input_char();
state=0;     // starting state
while (char != EOF) {
     state ← table(state,char);
     if (state == '-') return failure;
     word=word+char;
     char=input_char();
}
if (state == FINAL) return acceptance; else return failure;
```

Lexical Analysis التحليل المعجمي

(مثال التعرف على أسماء المسجلات)

Register → r ((0|1|2) (Digit|ε) | (4|5|6|7|8|9) | (3|30|31))

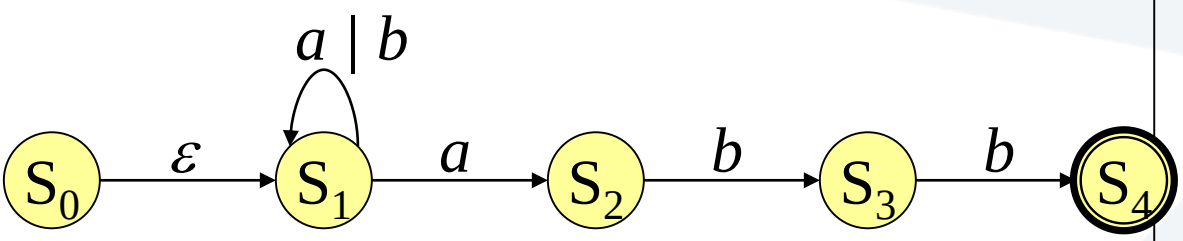


State	'r'	0,1	2	3	4,5,…,9
0	1	-	-	-	-
1	-	2	2	5	4
2(final)	-	3	3	3	3
3(final)	-	-	-	-	-
4(final)	-	-	-	-	-
5(final)	-	6	-	-	-
6(final)	-	-	-	-	-

- نفس كود المحاضرة السابقة.
- لكن مع جدول انتقال مختلف (أكبر من السابق).
- الآلية التحديدية المنتهية DFA هذه تميز المسجلات r_0-r_{31} فقط.
- لا يوجد انتقالان من نفس الحالة معنونان بنفس القيمة

الآلية غير التحديدية المنتهية
Non-deterministic Finite Automata (NFA)

*ماذا عن التعابير النظامية مثل (a | b)*abb*



- المثال يعبر عن آلية غير تحديدية منتهية
  - تمتلك S_0 انتقال معنوناً بـ ε، في حين تمتلك S_1 انتقالين معنونين بـ a (هذا غير ممكن في DFA).
- يعتبر DFA حالة خاصة من NFA.
  - يوجد على الأكثر قاعدة واحدة لكل حالة ولكل انتقال.
- يمكن مكافأة مخطط DFA بمخطط NFA (بسهولة).
- يمكن مكافأة مخطط NFA بمخطط DFA (بسهولة أقل).
  - من خلال مكافأة مجموعات من الحالات الممكنة.

لماذا ندرس آلية NFA؟

1- صحيح أن آلية DFA تقود إلى مميزات Recognizer أسرع من آلية NFA لكنها قد تكون أكبر حجماً.

2- تعتبر عملية تحويل التعابير النظامية RE إلى آلية NFA أكثر سهولة ومباشرة من التحويل إلى DFA.

التحليل المعجمي Lexical Analysis

لتحويل الوصف إلى شيفرة:

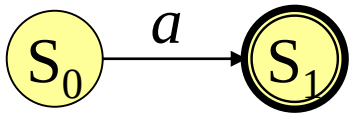
- كتابة التعابير النظامية التي تصف اللغة.
- التحويل من تعابير نظامية إلى آلية NFA (طريقة طومسون Thompson's construction).
- بناء آلية DFA التي تحاكي NFA (طريقة بناء المجموعات الجزئية).
- تقليص أو اختصار DFA (خوارزمية هوب كروفت Hopcroft's algorithm)

مولدات المحللات المعجمية:

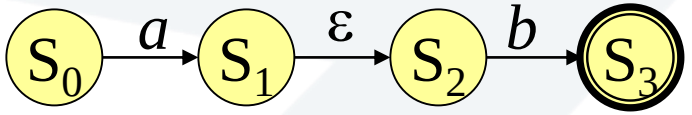
- أدوات برمجية مثل lex أو flex تقوم بكامل المهمة.
- المسألة الأساسية هي واجهة المعرب parser.

Lexical Analysis التحليل المعجمي

الفكرة الأساسية من توليد نماذج آلية NFA لكل رمز و/أو مشغل هي: ربطهم وفق ترتيب أولوياتهم. في الآتي بعض القواعد الأساسية لعمليات التحويل:



NFA for *a*



NFA for *ab*

عدد الحالات=

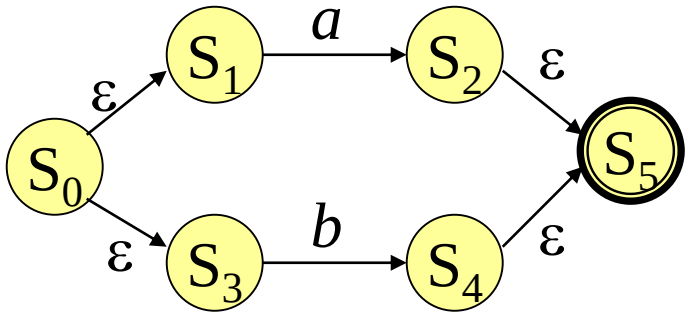
2s

S: عدد الرموز

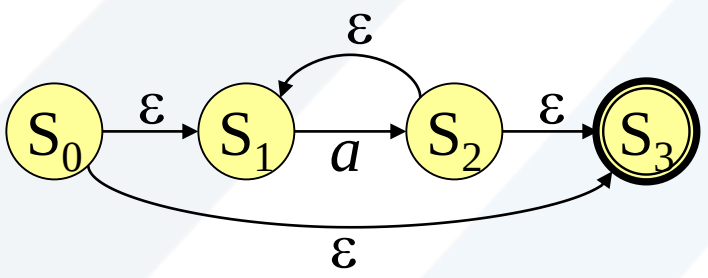
(بما فيها *، |، ε)

إضافة للرموز

a,b,..



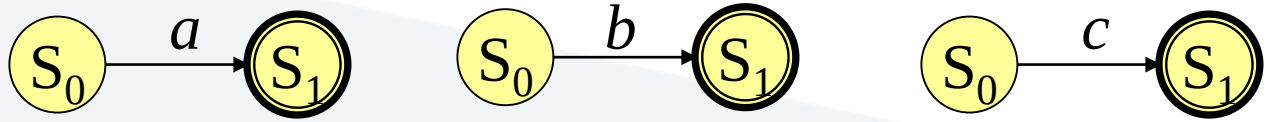
NFA for *a* | *b*

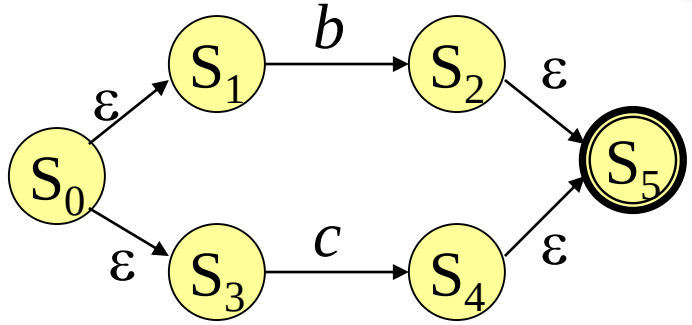


NFA for *a**

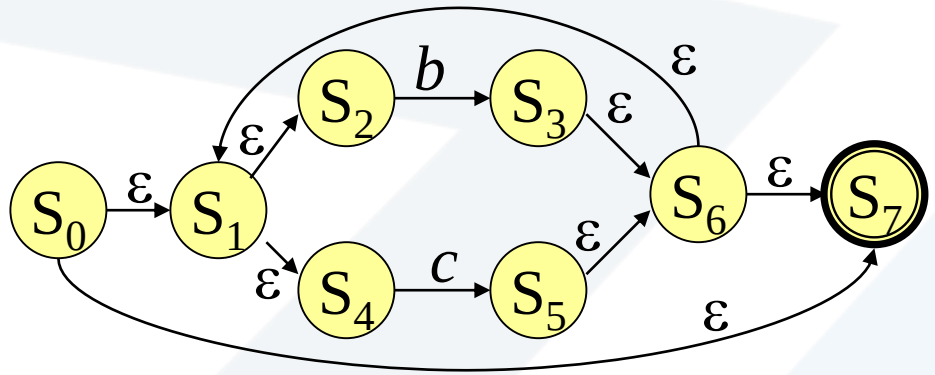
Lexical Analysis التحليل المعجمي

1- بداية نبني NFA للتعابير *a, b, c*



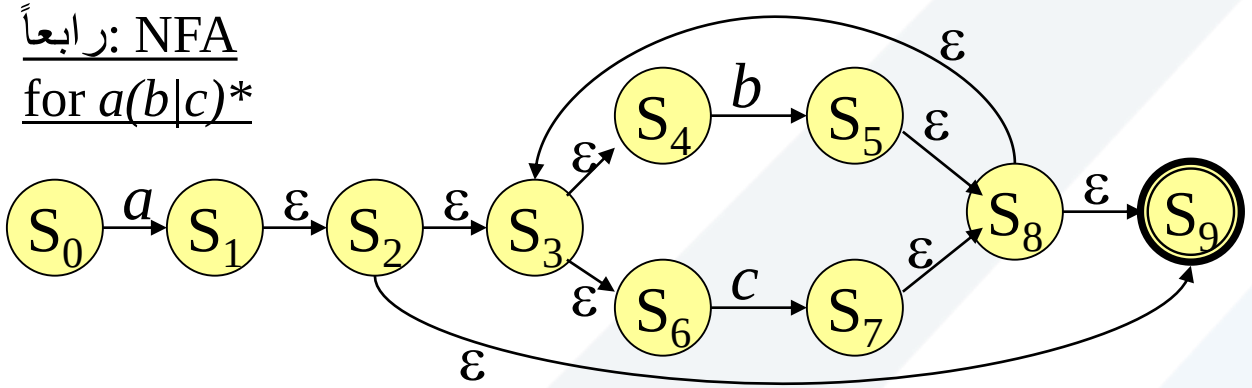


ثانياً: NFA for *b|c*

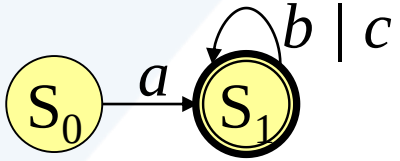


ثالثاً: NFA for *(b|c)**

رابعاً: NFA for *a(b|c)**



• ربما يقوم المصمم بوضع تصميم أبسط إلا أننا باتباع الخطوات السابقة نستطيع أتمتة إنتاج تصميم لبنية أكثر تعقيداً.



مثال بناء آلية NFA للتعبير النظامي:

$a\,(b|c)^*$

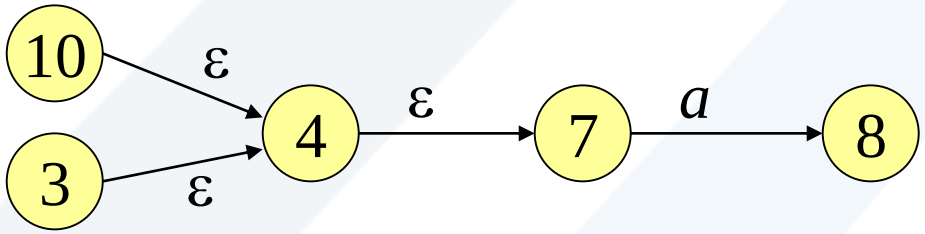
التحويل من NFA إلى DFA: عمليتان أساسيتان

(طريقة بناء المجموعات الجزئية)

- **move(s_i,a): (اجتماع)** مجموعة الحالات التي يوجد إليها انتقال من الحالة s_i معنون بالرمز a.
- **أداة الإنهاء ε-closure(s_i): (اجتماع)** مجموعة الحالات التي يمكن الوصول إليها انطلاقاً من الحالة s_i باتباع الانتقال المعنون برمز ε.

مثال:

- ε-closure(3)={3,4,7}; ε-closure({3,10})={3,4,7,10};
- move(ε-closure({3,10}),*a*)=8;



التحويل من NFA إلى DFA: عمليتان أساسيتان
(طريقة بناء المجموعات الجزئية)

- **المدخلات**: NFA ونختصرها بـ N.
- **المخرجات**: DFA وتختصر بـ D.
- **بعض المصطلحات الهامة:**
  - DTran : يمثل الجدول الانتقالي لـ DFA.
  - Dstate : يمثل مجموعة الحالات في الـ DFA

بدايةً، ε-closure هي الحالة الوحيدة الموجودة في Dstates وحالتها غير معلمة unmarked.

طالما لا زال هناك حالة T غير معنونة unmarked نفذ الخطوات التالية:

1- عنون T.

2- من أجل كل رمز دخل a نحسب المجموعة U

- حيث U تساوي أو تقابل مجموعة الحالات التي يتم الانتقال إليها من مجموعة الحالات T بالرمز المدخل a ونعبر عنها بـ

U:=ε-closure(move(T,a))

- **في حال كانت** U غير موجودة في Dstates

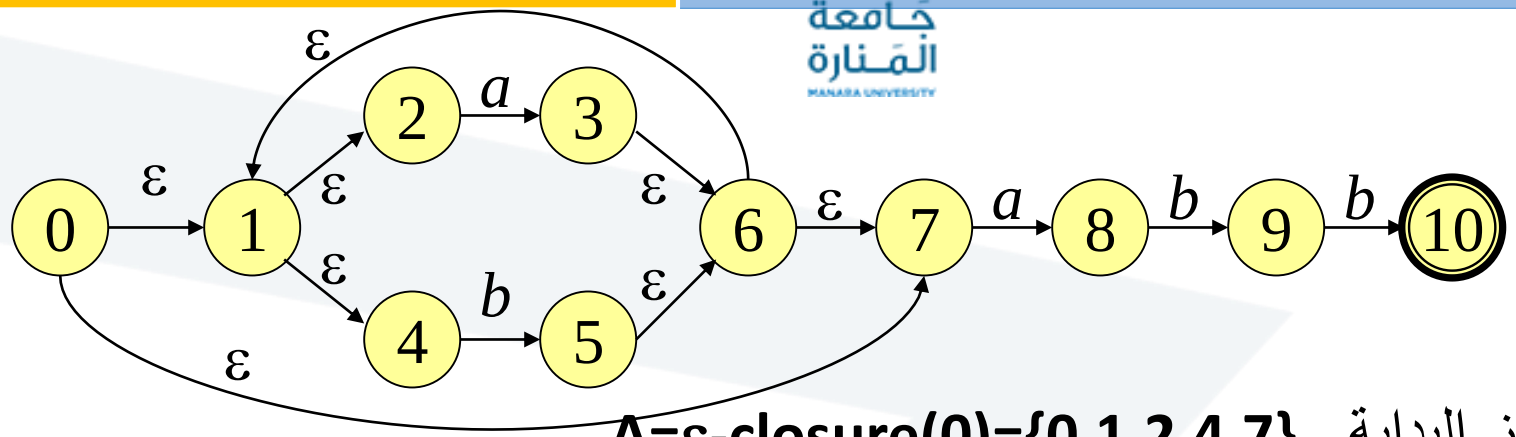
قم بإضافة U كحالة غير معنونة unmarked للحالات Dstates

حدث الجدول Dtable : Dtable[T,a]:=U

نتيجة: كل حالة من حالات DFA ترتبط بمجموعة من حالات NFA التي يمكن أن تأخذها بعد قراءة تتالٍ ما لرموز الدخل.

- تعتبر هذه ضمن حسابات الفاصلة الثابتة. fixed-point computation

التحليل المعجمي Lexical Analysis



- حساب المجموعة الأولى لرمز البداية **A=ε-closure(0)={0,1,2,4,7}**
- من أجل رمزي الدخل a,b سنقوم بالخطوات التالية:

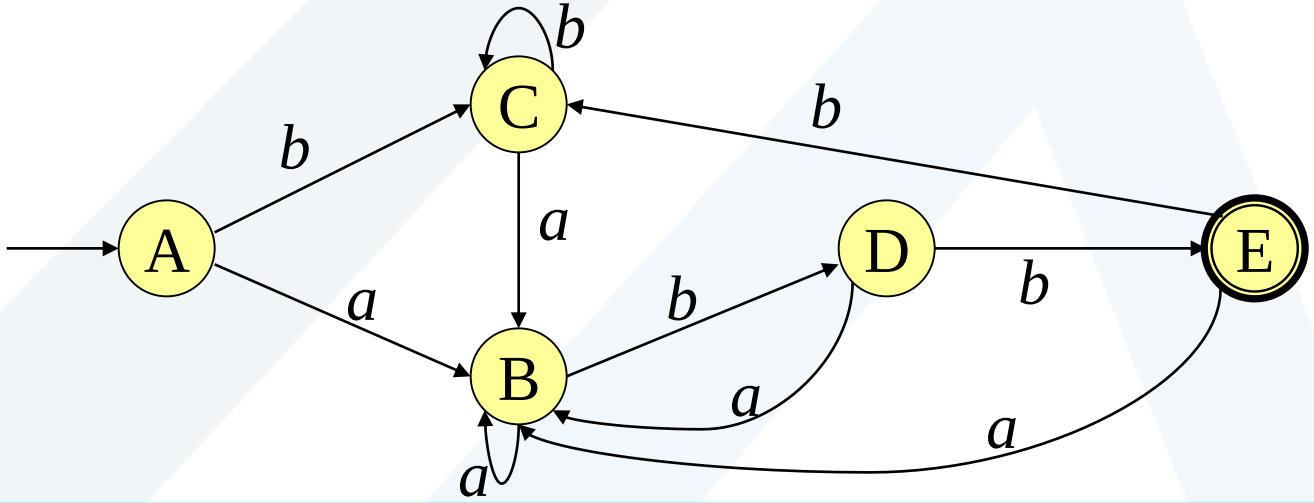
- ε-closure(move(A,*a*))=ε-closure({3,8})={1,2,3,4,6,7,8} → Dtable[A,*a*]=B
- ε-closure(move(A,*b*))=ε-closure({5})={1,2,4,5,6,7} → Dtable[A,*b*]=C
- ε-closure(move(B,a))=ε-closure({3,8})={1,2,4,5,6,7} =B → Dtable[B,a] = B
- ε-closure(move(B,b))=ε-closure({5,9})={1,2,4,5,6,7,9} =**D (new)** → Dtable[B,b] = D
- ε-closure(move(C,a))=ε-closure({3,8})={1,2,3,4,6,7,8} =B → Dtable[C,a] = B
- ε-closure(move(C,b))=ε-closure({5})={1,2,4,5,6,7} =C → Dtable[C,b] = C
- ε-closure(move(D,a))=ε-closure({3,8})={1,2,3,4,6,7,8} =B → Dtable[D,a] = B
- ε-closure(move(D,b))=ε-closure({5,10})={1,2,4,5,6,7,10} =**E (New)** → Dtable[D,b] = E
- ε-closure(move(E,a))=ε-closure({3,8})={1,2,3,4,6,7,8} =B → Dtable[E,a] = B
- ε-closure(move(E,b))=ε-closure({5})={1,2,4,5,6,7} =C → Dtable[E,b] = C

التحويل من NFA إلى DFA: عمليتان أساسيتان

(طريقة بناء المجموعات الجزئية)

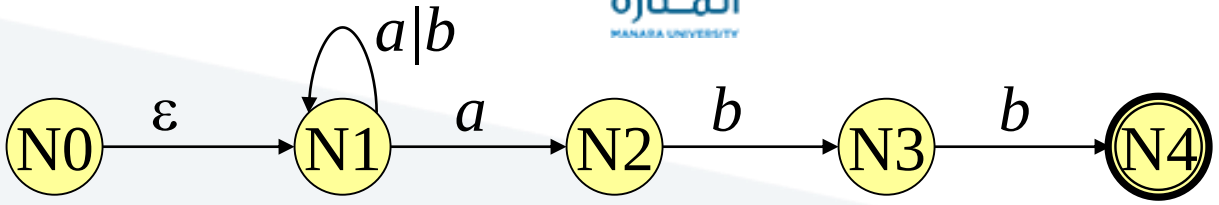
Transition table: جدول الانتقال

state	*a*	*b*
A	B	C
B	B	D
C	B	C
D	B	E
E(final)	B	C



التحويل من NFA إلى DFA: عمليتان أساسيتان

(طريقة بناء المجموعات الجزئية)



تطبيق خوارزمية بناء المجموعات الجزئية على مخطط NFA السابق:

Iteration	State	Contains	ε-closure(move(s,*a*))	ε-closure(move(s,*b*))
0	A	N0,N1	N1,N2	N1
1	B	N1,N2	N1,N2	N1,N3
	C	N1	N1,N2	N1
2	D	N1,N3	N1,N2	N1,N4
3	E	N1,N4	N1,N2	N1

ملاحظة:

- التكرار رقم 3 لا يضيف أي شيء لذلك نتوقف.
- الحالة E تتضمن الحالة النهائية N4.

التحليل المعجمي Lexical Analysis

- قدمنا خوارزمية لتحويل التعابير النظامية إلى مخططات DFA وهي خوارزمية بناء المجموعات الجزئية.
- ليس بالضرورة أن يكون مخطط DFA هو الحالة الأكثر اختصاراً (الأصغر).
- يمكننا بناء محلل معجمي آلي من التعابير النظامية باستخدام جدول الانتقال والكود اللازم لعملية النقل لكن يمكن أن يكون حجم الجدول كبيراً ! لذلك حولنا إلى DFA.
- Aho2 Sections 3.6-3.7; Aho1 pp. 113-125; Grune 2.1.6.1-2.1.6.6 (different style); Hunter 3.3 (very condensed); Cooper1 2.4-2.4.3