

الآليات المحدودة

FINITE AUTOMATA

FINITE AUTOMATON آلية محدودة



- كيف يمكن أن نقرر بأن نص ما string هو أحد المفردات المقبولة؟
- يستخدم برامج تعرف بالآليات المحدودة يمكنها التمييز بين أنماط النصوص.
- تسمى بالآلات/آليات الحالات المحدودة finite state machines
- وتسمى Recognizers أو Acceptors
- أي تميز النص أو تقبل/توافق على النص



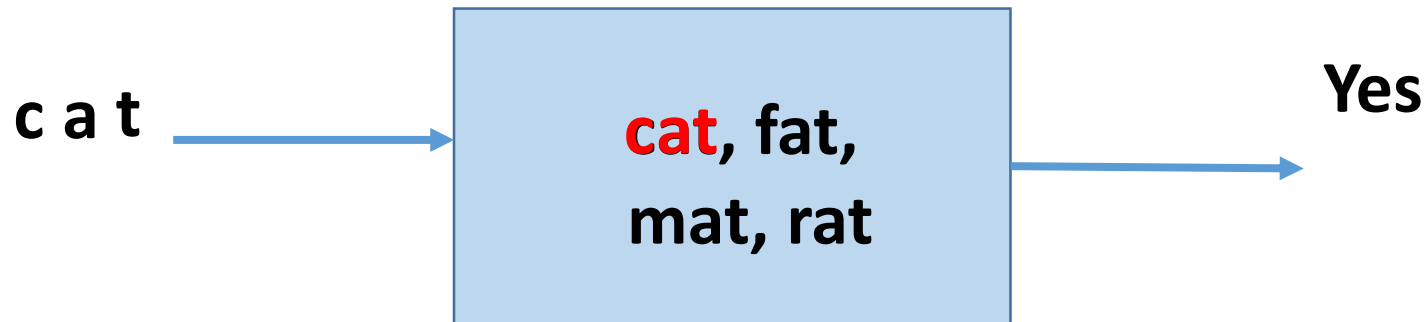
- المميز Recognizer أو Acceptor الخاص بلغة ما هو برنامج مدخلاته هي نص text (ولنرمز له بالمتغير str) والمميز يجيب:
 - ✓ "نعم" إذا كانت str مفردة lexem في اللغة
 - "لا" غير ذلك
- ويمكننا متابعة سلوك المميز من خلال بناء مخطط انتقالي transitional diagram
يسمى finite automaton



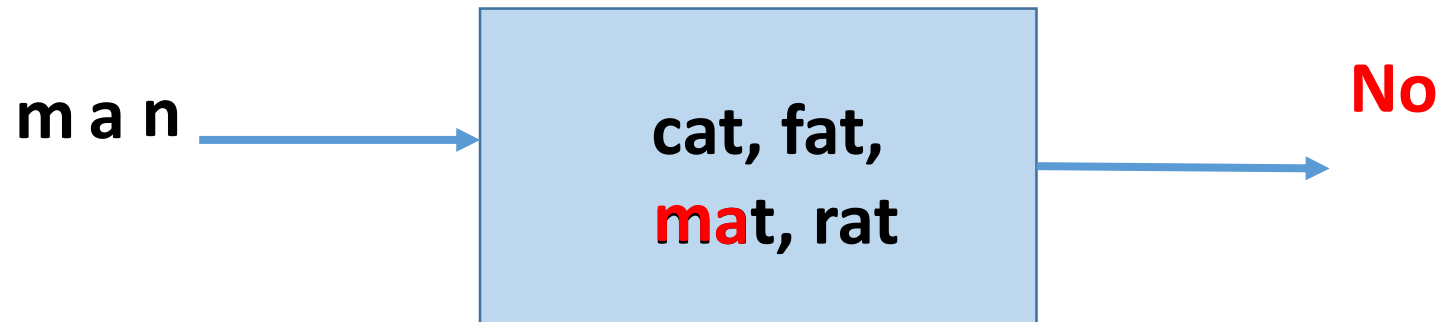
آلية عمل المميز (الآلية المحدودة)



آلية عمل المميز (الآلية المحدودة)

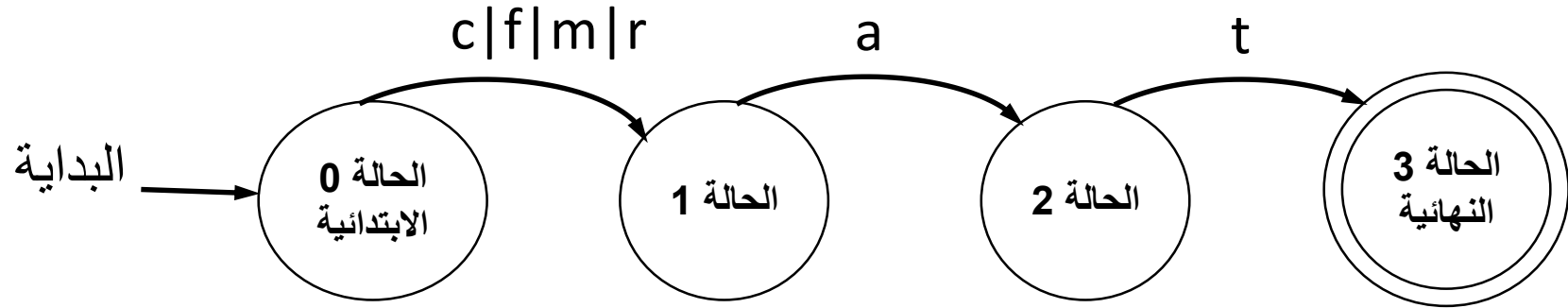


آلية عمل المميز (الآلية المحدودة)



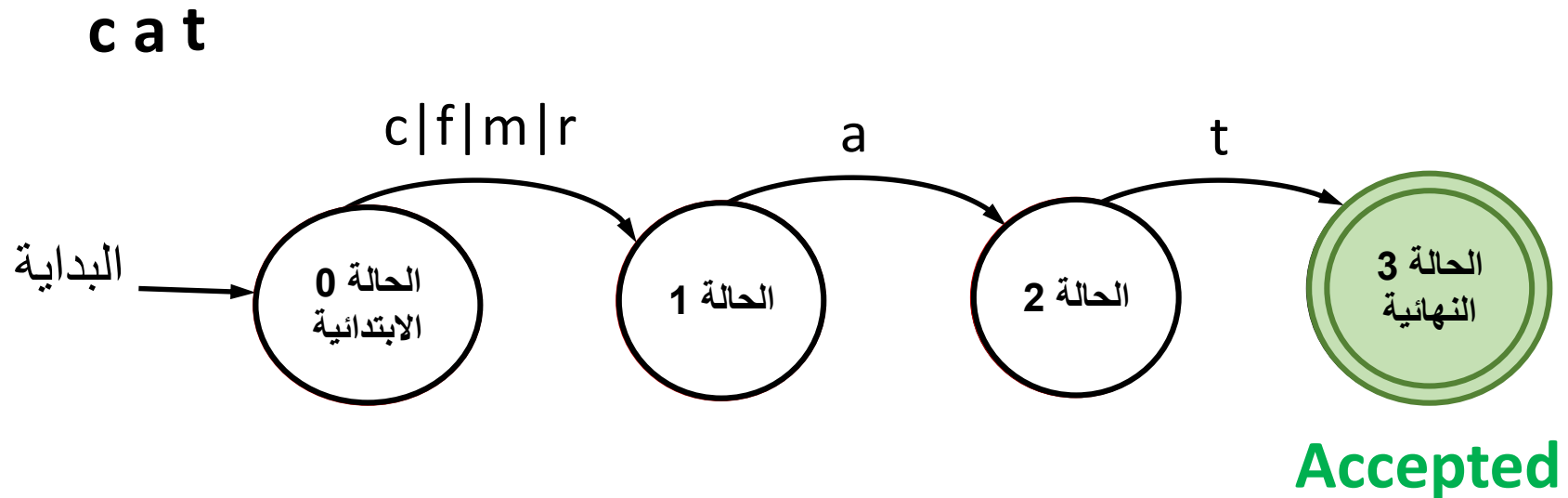
فكرة آلية محددة

مخطط لآلية تتعرف على نمط لغة التعبير النظامي:



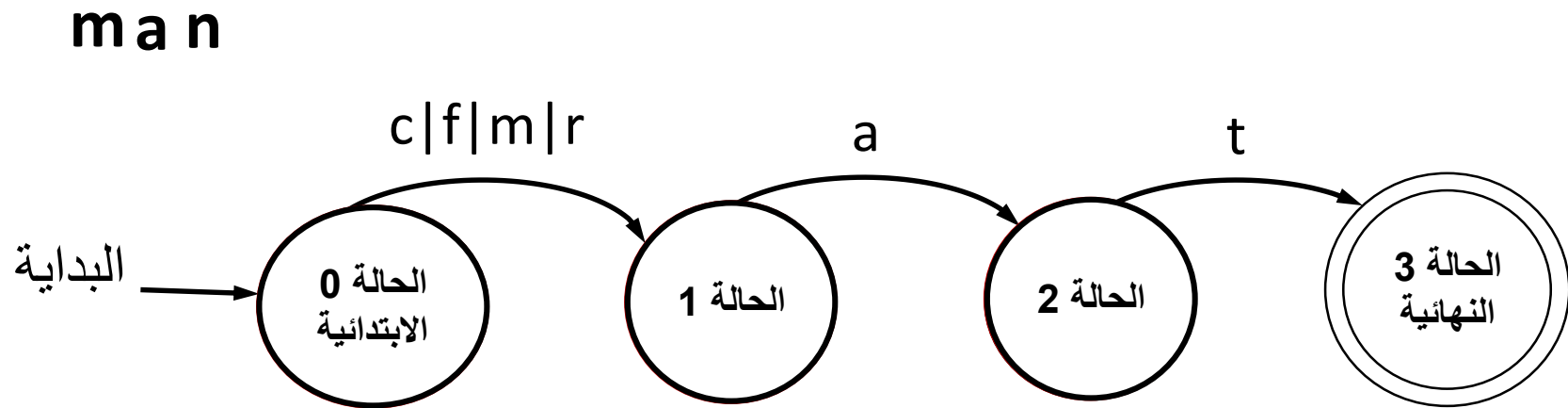
فكرة آلية محددة

مخطط لآلية تتعرف على نمط لغة التعبير النظامي:



فكرة آلية محددة

مخطط لآلية تتعرف على نمط لغة التعبير النظامي:



Not Accepted



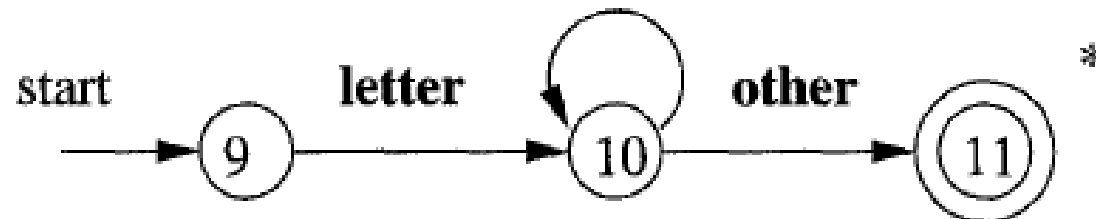
TRANSITIONAL DIAGRAM

المخطط الانتقالي

- كخطوة أولية في بناء محلل النص Lexical Analyzer, نقوم بتحويل نمط النص إلى مخطط انسيابي يسمى بالمخطط الانتقالي transitional diagrams
- وهو يحتوي على عقد/دوائر تسمى بالحالات states كل منها تمثل حالة يمكن أن تحدث أثناء عملية تمشيظ/مسح المدخلات بحثاً عن المفردات التي تطابق أنماطاً معينة.

letter \rightarrow [a – z]

digit \rightarrow [0 – 9]

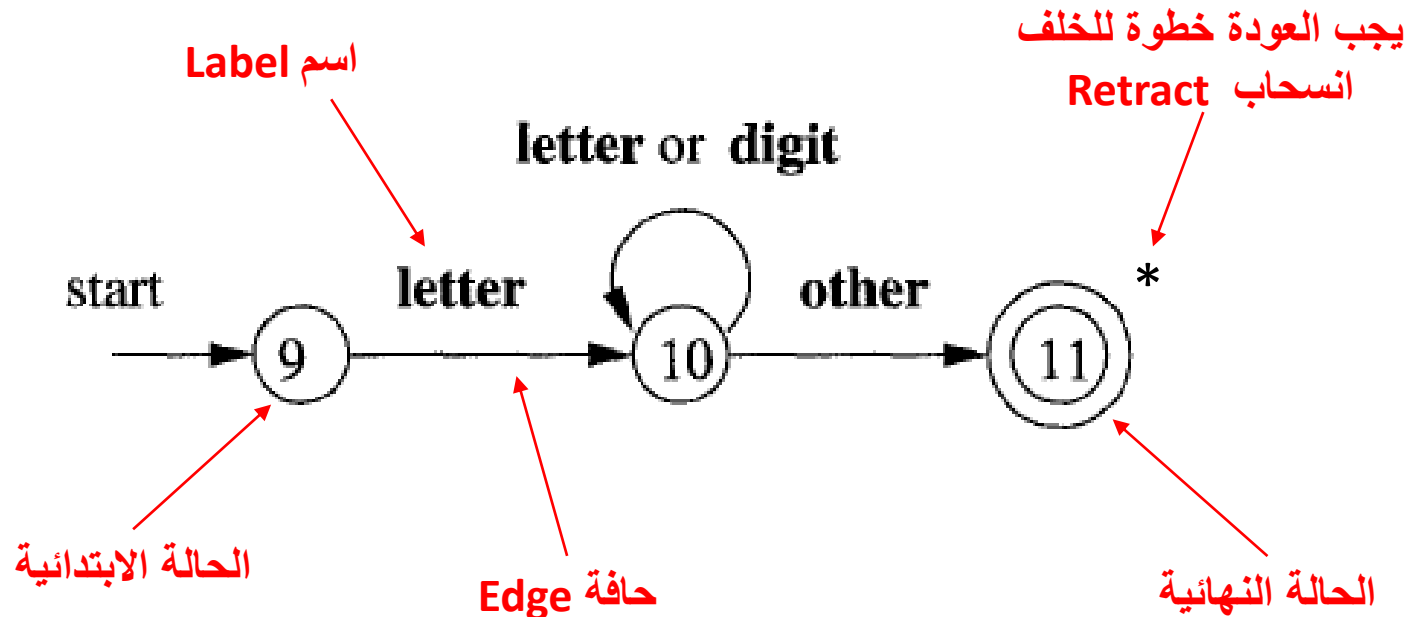


مخطط انتقالي للتعرف على نمط المعرفات



TRANSITIONAL DIAGRAM

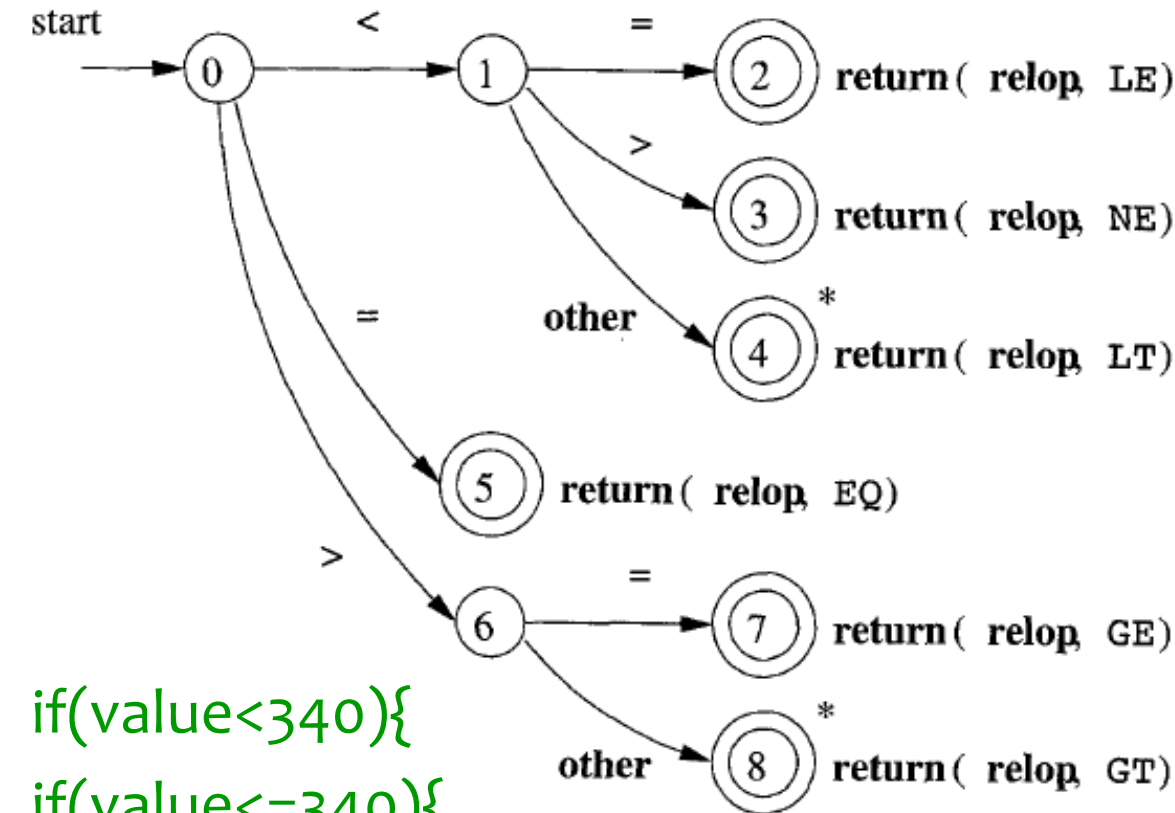
المخطط الانتقالي



- إذا نحن في حالة ما s وجاء مدخل ما i , عندها نبحث عن حافة خارجة من الحالة s عليها اسم label بالمدخل i ثم نتبع اتجاه سهم تلك الحافة لتوصلنا للحالة التالية t

TRANSITIONAL DIAGRAM

مثال للمخطط الانتقالي



```

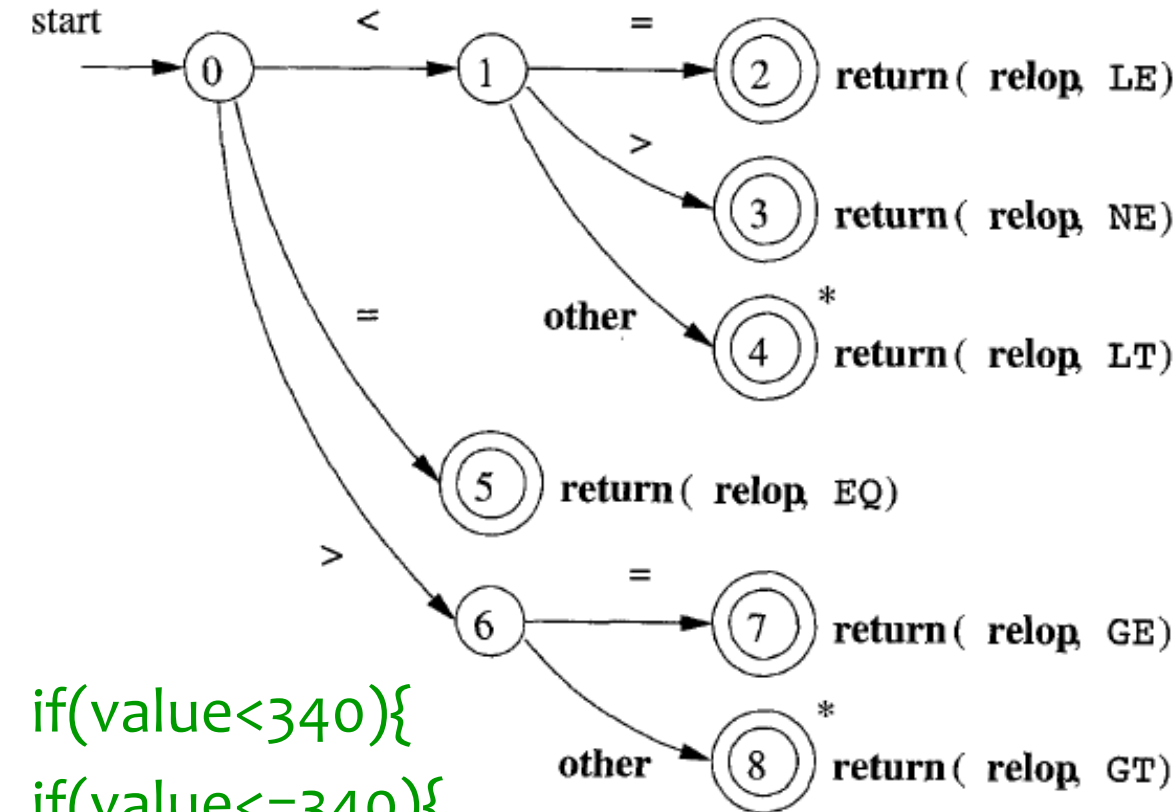
if(value<340){
if(value<=340){
if(value<>340){
if(value=340){

```

- نبدأ من حالة البداية 0
- إذا كان المدخل الأول هو الرمز '<' فننتقل للحالة التالية (1)، ولنقرر هل المدخل هو من البطاقة token "relop" فعلياً النظر للأمام forward لنقرر ذلك.
- إذا كان المدخل التالي هو الرمز '=' فنكون وصلنا حالة قبول/نهاية (2) ونتحصل على المفردة LE المعرفة من البطاقة token "relop"



TRANSITIONAL DIAGRAM



```

if(value<340){
if(value<=340){
if(value<>340){
if(value=340){

```

مثال للمخطط الانتقالي

- أما إذا كان المدخل التالي هو الرمز $>$ فنكون وصلنا حالة قبول/نهاية (3) ونتحصل على المفردة NE المعرفة من البطاقة token "relop"
- أما إذا كان الرمز التالي أي شيء آخر other بالتالي تحصلنا على المفردة LT من البطاقة "relop" ويتوجب رجوع مؤشر سلسلة النص للخلف retract
- إذا كان المدخل الأول هو الرمز $=$ فإننا تحصلنا على المفردة EQ

وهكذا مع $>$ و $=$ و other

بناء خوارزمية مخطط انتقالي

TOKEN getRelop() // TOKEN has two components: name and value

{

TOKEN retToken = new("RELOP"); // First component set here

while (true) {

switch(state) {

case 0: c = nextChar();

if (c == '<') state = 1;

else if (c == '=') state = 5;

else if (c == '>') state = 6;

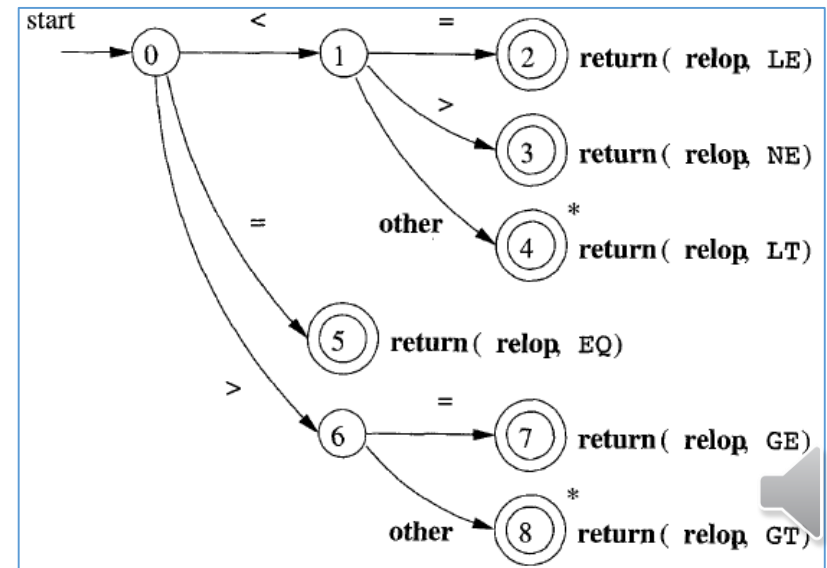
else fail();

break;

case 1: ...

}}}

if(value<340){



بناء خوارزمية مخطط انتقالي

TOKEN getRelop() // TOKEN has two components: name and value

{

TOKEN retToken = new("RELOP"); // First component set here

while (true) {

switch(state) {

case 0: c = **nextChar**();

if (c == '<') state = 1;

else if (c == '=') state = 5;

else if (c == '>') state = 6;

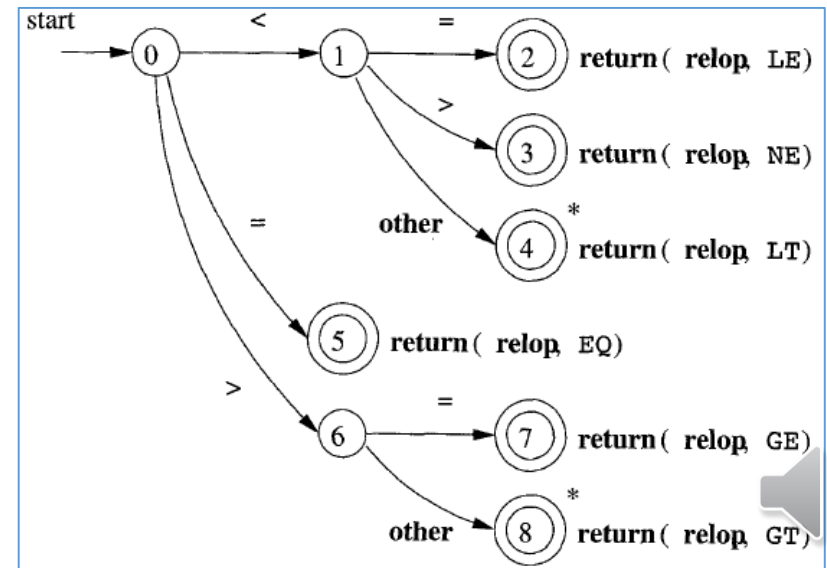
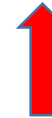
else **fail**();

break;

case 1: ...

}}}

if(value<340){



بناء خوارزمية مخطط انتقالي

```
TOKEN getRelop( ) // TOKEN has two components: name and value
{
```

```
    TOKEN retToken = new("RELOP"); // First component set here
```

```
    while (true) {
```

```
        switch(state) {
```

```
            case 0: ...
```

```
            case 1: c = nextChar( );
```

```
                if (c == '=') state = 2;
```

```
                else if (c == '>') state = 3;
```

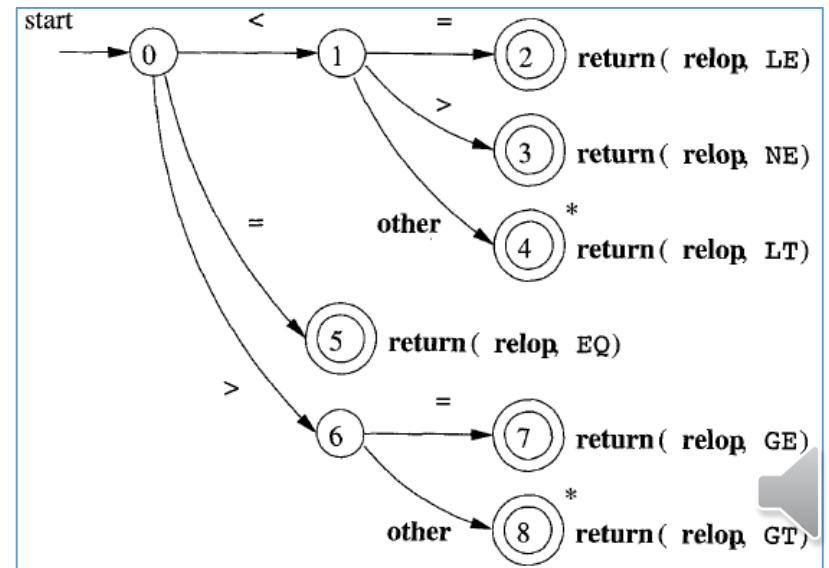
```
                else state = 4;
```

```
            break;
```

```
        ...
```

```
    } } }
```

if(value<340){



بناء خوارزمية مخطط انتقالي

```
TOKEN getRelop( ) // TOKEN has two components: name and value
{
```

```
    TOKEN retToken = new("RELOP"); // First component set here
```

```
    while (true) {
```

```
        switch(state) {
```

```
            case 0: ...
```

```
            case 1: c = nextChar( );
```

```
                if (c == '=') state = 2;
```

```
                else if (c == '>') state = 3;
```

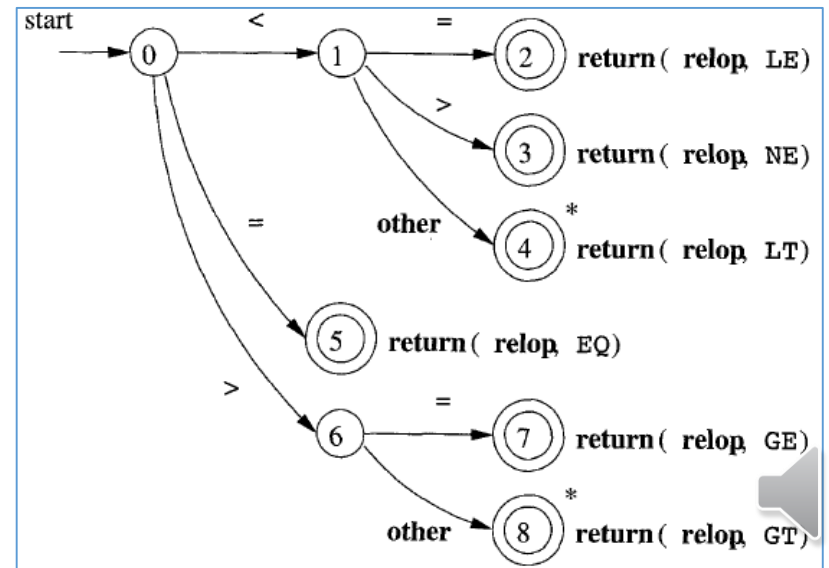
```
                else state = 4;
```

```
            break;
```

```
        ...
```

```
    } } }
```

if(value<340){



بناء خوارزمية مخطط انتقالي

TOKEN getRelop() // TOKEN has two components: name and value

{

TOKEN retToken = new("RELOP"); // First component set here

while (true) {

switch(state) {

case 0: ...

case 1: ...

case 2: ...

case 3: ...

case 4: retToken.attribute = "LT"; // second component is set here

retract(); // an accepting state with a star

return(retToken);

case 5: ...

case 6: ...

case 7: ...

case 8: ...

}}}

if(value<340){



بناء خوارزمية مخطط انتقالي

TOKEN getRelop() // TOKEN has two components: name and value

{

TOKEN retToken = new(“RELOP”); // First component set here

while (true) {

switch(state) {

case 0: ...

case 1: ...

case 2: ...

case 3: ...

case 4: retToken.attribute = “LT”; // second component is set here

retract(); // an accepting state with a star

return(retToken);

case 5: ...

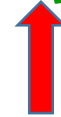
case 6: ...

case 7: ...

case 8: ...

}}}

if(value<340){



بناء خوارزمية مخطط انتقالي

- خوارزمية الدالة/الوظيفة `getRelop()` وظيفتها محاكاة مخطط relop الانتقالي وتعود `returns` بكائن من نوع `TOKEN`
- الكائن `TOKEN` يتكون من عنصرين وهما اسم البطاقة, وقيمة صفة البطاقة.
- بدايةً تكون الوظيفة `getRelop()` كائن جديد `retToken` وتجهزه بالعنصر الأول (اسم البطاقة) `RELOP`
- الحالة الابتدائية 0 ممثلة في التعليمة `case 0:`
- الدالة `nextChar()` تجلب الرمز التالي من سلسلة المدخلات وتسلمها للمتغير `c`
- بعدها نختبر `c` حول الرموز الثلاثة المتوقعة (`>`, `<`, `=`) لننتقل بين الحالات



بناء خوارزمية مخطط انتقالي

- إذا كان الرمز المدخل التالي ليس من ضمن رموز الاختبارات, عندها تستدعي الدالة **fail()** لتوقف قراءة الرموز والعودة إلى **بداية سلسلة رموز المفردة**
- **lexemeBegin** لكي يتولى مخطط انتقالي آخر التعامل مع السلسلة
- توضح الخوارزمية التصرف في الحالة 4 و 8 التي تحمل رمز النجمة *
- على الخوارزمية الانسحاب خطوة للخلف لمؤشر قراءة الرموز وذلك باستدعاء الدالة **retract()**
- بما أن الحالة 4 تمثل تمييز المفردة <, فعندها يتم وضع العنصر الثاني لكائن البطاقة (قيمة صفة token) لتكون في هذه الحالة LT



المحاضرة التالية:

نواصل الآلات المحدودة

CONTINUE FINITE AUTOMATA

