



1440

السنة الرابعة - نظم المعلومات وهندسة البرمجيات



12

19/11/2023

LR Parser & Ambiguous Grammar

د. باسم قصيبة



RB Informatics;

المترجمات

- تحدثنا في المحاضرة السابقة عن ال Bottom up parsing وتعرفنا عن بعض المفاهيم المهمة مثل Shift و Reduce وبعد ذلك قمنا بحل عدة تمارين عن ال LR Parser.
- أفكار هذه المحاضرة تعتمد بشكل أساسي على المحاضرة السابقة لذلك يجب التمكن من المحاضرة السابقة.
- في هذه المحاضرة سنراجع بعض أفكار المحاضرة السابقة بشكل سريع وبعد ذلك سنقوم بحل مثال شامل لأفكار المحاضرتين على بركة الله نبدأ.

1. الفرق بين ال parse tree و شجرة ال AST

شجرة ال AST هي بنية تم اختراعها لتمثيل الكود على شكل شجرة أما ال parse tree فهي عبارة عن جملة لاستدعاء التوابع بشكل عودي على سبيل المثال عندما يرى عبارة Reduce يقوم باستدعاء تابع ما و عندما يرى عبارة Shift يقوم باستدعاء تابع آخر.

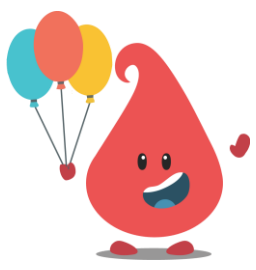
2. الفرق بين ال Shift و ال Reduce

ال Shift هي الانتقال الى حالة جديدة تحوي على القاعدة السابقة بعدما تم قراءة الدخل الحالي ونقله للمكدس (أي نقل المحرف التي وصلنا إليه من الدخل الى القمة من المكدس)

مثال:

state	stack	Input
Current state	\$ABC	XYZ\$
Next state	\$ABXC	YZ\$

ال Reduce هي استبدال قيم ال terminal بالقاعدة الموافقة له.



مثال :

لدينا القاعدة $A \rightarrow xy$

state	stack	input
Current state	\$Cbyx	ijk\$
Next state	\$CbA	ijk\$

3. خوارزمية إيجاد ال Action Table

For each state s_i and terminal a

- If s_i has item $X \rightarrow \alpha.ab$ and $\text{goto}[i,a] = j$ then $\text{action}[i,a] = \text{shift } j$
- If s_i has item $X \rightarrow \alpha.$ and $a \in \text{Follow}(X)$ then $\text{action}[i,a] = \text{reduce } X \rightarrow \alpha$
- If s_i has item $S' \rightarrow S.$ then $\text{action}[i,\$] = \text{accept}$
- Otherwise, $\text{action}[i,a] = \text{error}$

شرح الخوارزمية:

من أجل كل حالة s_i وال terminal a فإن:

- عندما نكون في الحالة i والقاعدة الحالية $X \rightarrow \alpha.ab$ (النقطة تعبر أن المحرف التالي التي سيتم إدخاله للمكدس هو المحرف التي يتبع النقطة في مثالنا هو a) حيث $\text{goto}[i,a] = j$ أي عندما نكون في الحالة i وقمنا بقراءة المحرف a سوف نتقل لحالة جديدة j وعندئذ يكون $\text{action}[i,a] = \text{shift } j$ أي نقل ال a الى المكدس ويصبح لدينا $X \rightarrow \alpha.a.b$ ونحن في حالة جديدة j
- عندما نكون في الحالة i والقاعدة الحالية $X \rightarrow \alpha.$ (النقطة في النهاية أي لم يعد هناك شيء يمكن إدخاله للمكدس) أي تمت قراءة القاعدة بالكامل ومع ذلك لم نصل الى ال start symbol وبالتالي في هذه الحالة سنقوم بحساب ال $\text{Follow}(X)$ وبفرض أن قيمه تساوي ال a حيث $a \in \text{Follow}(X)$ وبعد ذلك نقوم بحساب الانتقالات عند هذه القيم أي $\text{action}[i,a] = \text{reduce } X \rightarrow \alpha$ (استبدال جميع القيم المقابلة للقاعدة $X \rightarrow \alpha$ في الحالة i)
- عندما نكون في الحالة i والقاعدة الحالية $S' \rightarrow S.$ (أي تم الوصول الى ال start symbol وقراءته بشكل كامل وكانت قيمة الدخل التالي تساوي ل $\$$)
- في هذه الحالة تكون القاعدة مقبولة أي $\text{action}[i,\$] = \text{accept}$
- أي انتقال آخر يكون عبارة عن خطأ وسنعالج هذه الحالة في المحاضرة القادمة



شرح
الفقرة السابقة

4. خوارزمية عمل ال LR Parsing

```

let a be the first symbol of w$;
while(1) { /* repeat forever/*
let S be the state on top of the stack;
if ( ACTION[S, a] = shift t ){
    ■ push t onto the stack;
    ■ let a be the next input symbol;
}
else if ( ACTION[S, a] = reduce A → β ){
    ■ pop |β| symbols of the stack;
    ■ let state t now be on top of the stack;
    ■ push GOTO[t, A] onto the stack;
    ■ output the production A → β
}
else if ( ACTION[S, a] = accept ) break; /* parsing is done/*
else call error-recovery routine;
}

```

مثال:

بفرض لدينا القواعد التالية:

$$E \rightarrow E+E \mid E * E \mid (E) \mid id$$

والمطلوب:

1- إيجاد جدول ال LR

2- التحقق من الدخل $id * id + id$ اذا كان يحقق القواعد بالاعتماد على الجدول السابق

الحل:

في البداية نضيف قاعدة جديدة قيمتها تساوي start symbol وتعتبر هي ال start symbol الجديد ($S \rightarrow E$) وبعد ذلك نقوم بنشر القواعد (أي وضع كل بديل كقاعدة منفصلة) و ثم نرقمها جميعها باستثناء القاعدة الجديدة.

فيكون لدينا القواعد التالية:

$S \rightarrow E$
 $E \rightarrow E+E$ (1)
 $E \rightarrow E * E$ (2)
 $E \rightarrow (E)$ (3)
 $E \rightarrow id$ (4)

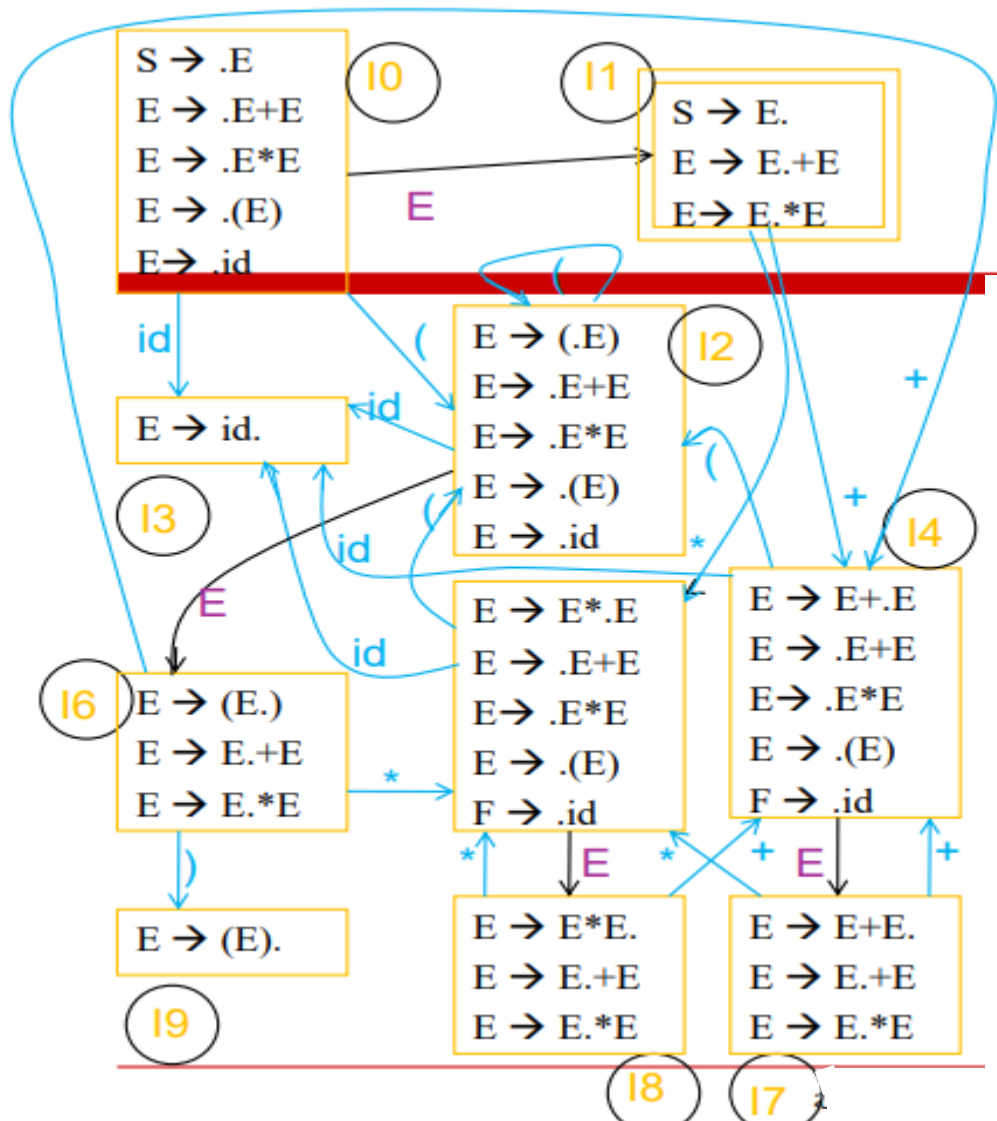


حساب ال First & Follow للقواعد السابقة:

$\text{First}(E) = \{ id, (\}$

$\text{Follow}(E) = \{ \$, +, *,) \}$

والآن نقوم بإيجاد الحالات انطلاقاً من ال start symbol



شرح كيف تم إيجاد الحالات السابقة:

شرح الحالة الأولى 10 :

$$\begin{aligned}
 S &\rightarrow .E \\
 E &\rightarrow .E+E \\
 E &\rightarrow .E * E \\
 E &\rightarrow .(E) \\
 E &\rightarrow .id
 \end{aligned}$$


كما نعلم نبدأ قراءة الدخّل من ال start symbol (أي نضع نقطة في البداية من جميع بدائل S) لذلك $S \rightarrow .E$ حيث النقطة تعبر عن أن المكّس قد وصل بقراءة الدخّل لهذا المكان (مكان النقطة) والآن سيقوم بقراءة ما بعد النقطة وبالتالي سيقرأ E لأن القاعدة $S \rightarrow .E$ وبما أنه سيبدأ بقراءة E إذا نقوم بوضع نقطة في البداية من جميع بدائل E أي :

$$\begin{aligned}
 E &\rightarrow .E+E \\
 E &\rightarrow .E * E \\
 E &\rightarrow .(E) \\
 E &\rightarrow .id
 \end{aligned}$$

ومن الحالة السابقة نلاحظ الحالات التالية سوف نصل إليها بقراءة E (أو id) :

$$\begin{aligned}
 S &\rightarrow .E \\
 E &\rightarrow .E+E \\
 E &\rightarrow .E * E
 \end{aligned}$$

الحالة الثانية بقراءة () وذلك عندما تكون القاعدة :

$$E \rightarrow .(E)$$

الحالة الثالثة بقراءة id وذلك عندما تكون القاعدة :

$$E \rightarrow .id$$

وبالتالي :

1. عندما يقرأ E سوف ينتقل للحالة 11
2. عندما يقرأ (سوف ينتقل للحالة 12
3. عندما يقرأ id سوف ينتقل للحالة 13

والآن ننتقل لشرح الحالة الثانية 11

الحالة الأولى I0

$$\begin{aligned}
 S &\rightarrow .E \\
 E &\rightarrow .E+E \\
 E &\rightarrow .E*E \\
 E &\rightarrow .(E) \\
 E &\rightarrow .id
 \end{aligned}$$

قراءة E

الحالة الثانية I1

$$\begin{aligned}
 S &\rightarrow E. \\
 E &\rightarrow E.+E \\
 E &\rightarrow E.*E
 \end{aligned}$$

شرح الحالة الثانية I1 :

ومن الحالة السابقة نلاحظ الحالات التالية سوف نصل اليها بقراءة + أو * :

■ الحالة الأولى عندما تكون القاعدة $(S \rightarrow E.)$:

نلاحظ ان هذه القاعدة هي نفسها ال start symbol وأن المحرف التالي التي سيتم قراءته هو \$ وذلك

لأننا وصلنا لنهاية القاعدة وبالتالي هذه الحالة مقبولة

■ الحالة الثانية بقراءة + وذلك عندما تكون القاعدة:

$$E \rightarrow E.+E$$

■ الحالة الثالثة بقراءة * وذلك عندما تكون القاعدة:

$$E \rightarrow E.*E$$

وبالتالي :

1. عندما يقرأ + سوف ينتقل للحالة I4

2. عندما يقرأ * سوف ينتقل للحالة I5

والآن ننتقل لشرح الحالة الثالثة I2

شرح الحالة الثالثة I2 :

الحالة الأولى I0

$$\begin{aligned}
 S &\rightarrow .E \\
 E &\rightarrow .E+E \\
 E &\rightarrow .E*E \\
 E &\rightarrow .(E) \\
 E &\rightarrow .id
 \end{aligned}$$

(قراءة)

الحالة الثالثة I2

$$\begin{aligned}
 S &\rightarrow .(E) \\
 E &\rightarrow .E+E \\
 E &\rightarrow .E*E \\
 E &\rightarrow .(E) \\
 E &\rightarrow .id
 \end{aligned}$$
في هذه الحالة I0 نلاحظ عندما قمنا بقراءة (انتقلنا للحالة I2 والقاعدة التي قمنا بقراءتها هي $E \rightarrow .(E)$ وفي الحالة I2 كان لدينا $E \rightarrow .(E)$ وبالتالي وصلنا لنفس الحالة التي مرت معنا في الحالة I0 عند قراءة E

لذلك ايضاً سنقوم بقراءة جميع بدائل E في الحالة I2

ومن الحالة السابقة نلاحظ الحالات التالية سوف نصل اليها بقراءة E (أو id) :

■ الحالة الأولى بقراءة E وذلك عندما تكون القاعدة:

$$S \rightarrow (.E)$$

$$E \rightarrow .E+E$$

$$E \rightarrow .E * E$$

■ الحالة الثانية بقراءة $($ وذلك عندما تكون القاعدة:

$$E \rightarrow .(E)$$

■ الحالة الثالثة بقراءة id وذلك عندما تكون القاعدة:

$$E \rightarrow .id$$

وبالتالي :

1. عندما يقرأ E سوف ينتقل للحالة 16

2. عندما يقرأ $($ سوف يعود للحالة نفسها

3. عندما يقرأ id سوف ينتقل للحالة 13

والآن ننتقل لشرح الحالة الرابعة 13

شرح الحالة الرابعة 13 :

الحالة الأولى 10

$$\begin{array}{l} S \rightarrow .E \\ E \rightarrow .E+E \\ E \rightarrow .E * E \\ E \rightarrow .(E) \\ E \rightarrow .id \end{array}$$

قراءة id

الحالة الرابعة 13

$$E \rightarrow id.$$

الحالة الثالثة 12

$$\begin{array}{l} S \rightarrow (.E) \\ E \rightarrow .E+E \\ E \rightarrow .E * E \\ E \rightarrow .(E) \\ E \rightarrow .id \end{array}$$

قراءة id

الحالة الرابعة 13

$$E \rightarrow id.$$

ونلاحظ اننا وصلنا لنهاية الدخول في القاعدة $E \rightarrow id.$ وبالتالي لا يوجد حالات لهذه الحالة.

وبهذا الشكل نتابع إيجاد بقية الحالات

والآن بعد إيجاد جميع الحالات نقوم بإيجاد جدول ال LR و هو مقسوم لنصفين في البداية نوجد قسم ال Action وبعد ذلك قسم ال GoTo

ملاحظة:

قسم ال Action متعلق بال terminal أما قسم ال GoTo فهو متعلق بال non-terminal

	ACTION						GO
	+	*	()	id	\$	E
I0			S2		S3		1
I1	S4	S5				Acc	
I2			S2		S3		6
I3	R4	R4		R4		R4	
I4			S2		S3		7
I5			S2		S3		8
I6	S4	S5		S9			
I7	R1 S4	R1 S5		R1		R1	
I8	R2 S4	R2 S5		R2		R2	
I9	R3	R3		R3		R3	

شرح كيف تم ملئ قسم ال Action

- نلاحظ من الحالة I0 أنه لا يوجد انتقالات عند ال \$ أو) أو * أو + لذلك نترك خاناتها فارغة في القسم المقابل ل I0 وعند ال id أو (يوجد انتقال لحالة جديدة حيث عند ال (ينتقل للحالة I2 وبالتالي نقوم بعمل shift 2 أي انتقال للحالة I2 مع عمل shift فنلاحظ نقوم بملئ الخانة المقابلة ل (بالقيمة s2 وعند ال id ينتقل للحالة I3 وبالتالي نقوم بعمل shift 3 أي انتقال للحالة I3 مع عمل shift فنلاحظ نقوم بملئ الخانة المقابلة ل id بالقيمة s3 ونرمز للعملية السابقة اختصاراً $Action(0, id) = s3$

الحالة الحالية

الرمز المقروء

ال Action

- والآن نتقل للحالة I1 أنه يوجد انتقالات فقط عند ال \$ و * و + حيث:

$Action(1, +) = s4$, $Action(1, *) = s5$, $Action(1, \$) = Accept$

■ لماذا $Action(1, \$) = Accept$ ؟؟

نلاحظ في الحالة 1 في القاعدة $S \rightarrow E$ قد وصلنا لنهاية القاعدة أي الرمز التالي التي سيتم قراءته هو $\$$ والرمز S يعبر عن ال $start\ symbol$ لذلك الحالة مقبولة (راجع الخوارزمية اذا ما عرفت شو صار...)

■ والآن نتقل للحالة 12 أنه يوجد انتقالات فقط عند ال id و) حيث:

$Action(2, id) = s3$, $Action(2, () = s2$

■ والآن نتقل للحالة 13 حيث لا يوجد انتقالات ونلاحظ أنها تحوي على قاعدة تمت قراءتها بالكامل أي $E \rightarrow id$ (نقطة في نهاية القاعدة) في هذه الحالة نقوم بحساب ال $Follow$ للقاعدة أي $Follow(E)$ وملئ الحقول المقابلة لل $terminal$ التي تنتمي لنتاج $Follow(E)$ ب $Action(3, values\ Follow(E)) = Reduce\ (number\ of\ rule)$ وبالعودة للقواعد بعدما قمنا بترتيبها في بداية الصفحة 4 من محاضرتنا نلاحظ أن ترتيب القاعدة $E \rightarrow id$ هو 4 لذلك نملئ الحقول المقابلة أي ال $terminal$ التي تنتمي لنتاج $Follow(E)$ ب $R4$ أي $Reduce\ 4$

أي باختصار نوجد $Follow(E) = \{ \$, +, *,) \}$

وبعد ذلك نعوض $Action(3, ...) = Reduce\ 4$

وبنفس الخطوات نتابع بقية الحالات وصولاً للحالة 18 & 17

■ في السطر 18 & 17 من الجدول السابق في الخانة المقابلة لل $(+, *)\ terminal$ سوف يتم عمليتين في نفس الخانة وهذا غير صحيح لأنه سوف يسبب الغموض $Ambiguous$ لذلك علينا اختيار احد العمليتين حتى يكون الحل صحيح (ليست اختيارية عملية الاختيار وانما تتم بخطوات مدروسة كالتالي).

■ كيف تتم عملية الاختيار؟؟

■ في الحالة 17 نلاحظ أن الحالة التي وصلت للنهاية هي $E \rightarrow E + E$ لذلك عند ال $+$ تكون الأولوية لعملية الجمع التي أدت للوصول لنهاية احدى القواعد لذلك نقوم بعمل $Reduce\ E \rightarrow E + E$ وبالتالي نقوم باختيار $R1$ بينما عند ال $*$ تكون الأولوية للقاعدة التي تحوي عملية الضرب $E \rightarrow E * E$ لذلك نختار $Shift\ 5$

■ في الحالة 18 نلاحظ أن الحالة التي وصلت للنهاية هي $E \rightarrow E * E$ لذلك عند ال $*$ تكون الأولوية لها دائماً

- $Action(7,*) = S5$ not $R1$
 - State 7 = $E+E$. and we want shift "*" before reduce $E+E$
- $Action(7,+) = R1$ not $S4$
 - State 7 = $E+E$. and we want reduce not shift "+"
- $Action(8,*) = R2$ not $S5$
 - State 8 = $E * E$. and we want reduce $E * E$ before shift "*"
- $Action(8,+) = R2$ not $S4$
 - State 8 = $E * E$. and we want reduce $E * E$ before shift "+"

توضيح عملية الاختيار:

- إذا كان لدينا قاعدة وصلت لنهايتها وهي عملية ضرب فالأولوية دوماً لها
- إذا كان لدينا قاعدة وصلت لنهايتها وهي عملية جمع فالأولوية لها (مقابل terminal ال +) أما (مقابل terminal ال *) فالأولوية لعملية الضرب

🔗 شرح كيف تم ملئ قسم ال Goto

تحدثنا سابقاً أن ال Action يتعامل مع ال terminal بينما ال Goto يتعامل مع ال non-terminal لذلك نلاحظ أن الانتقالات عند ال non-terminal هي :

1. E عند الحالة 10
2. E عند الحالة 12
3. E عند الحالة 14
4. E عند الحالة 15

وبعد ذلك نقوم بحساب ال Action للحالات السابقة كالتالي:

الحالة التالية = $Action(non-terminal, \text{الحالة الحالية})$

مهمة التابع كالتالي :

"ماهي الحالة التي سأنتقل اليها عندما أكون في الحالة الحالية وال non-terminal هو ؟"

فيكون الحل :

$Action(0,E)=1$

$Action(2,E)=6$

$Action(4,E)=7$

$Action(5,E)=8$

■ الآن نريد التحقق اذا كان الدخل $id*id+id$ يحقق القواعد السابقة ؟؟

Stack	Input	Action
0	$id*id+id\$$	Shift 3
0 id 3	$*id+id\$$	Reduce $E \rightarrow id$
0 E 1	$*id+id\$$	Shift 5
0 E 1 * 5	$id+id\$$	Shift 3
0 E 1 * 5 id 3	$+id\$$	Reduce $E \rightarrow id$
0 E 1 * 5 E 8	$+id\$$	Reduce $E \rightarrow E * E$
0 E 1	$+id\$$	Shift 4
0 E 1 + 4	$id\$$	Shift 3
0 E 1 + 4 id 3	$\$$	Reduce $E \rightarrow id$
0 E 1 + 4 E 7	$\$$	Reduce $E \rightarrow E + E$
0 E 1	$\$$	Accept

■ شرح كيف تم ملئ الجدول ؟؟

بالاعتماد على جدول ال LR نقوم بملئ الجدول السابق حيث تتبع التالي:
 في البداية يكون المكسد يحوي الحالة البدائية أي الحالة 10 وبالنظر الى الدخل نلاحظ أن بدايته تساوي id فنقوم بالذهاب لجدول ال LR ونجد ال $Action(0,id)=S3$ وبالتالي يكون ال Action يساوي ل Shift 3 ونقوم بنقل ال id من الدخل الى المكسد والحالة الحالية التي وصلنا اليها هي 13 لذلك تصبح قيمة المكسد $0 id 3$ ومن قمة المكسد نلاحظ قيمته 3 وبالتالي نحن في الحالة 13 وقيمة الدخل الحالي هي * بالتالي يكون ال $Action(3,*)=R4$ نلاحظ انها Reduce بالتالي نحذف البداية من المكسد و نستبدل القيم في المكسد المقابلة للقاعدة رقم 4 أي $E \rightarrow id$ (نستبدل جميع قيم ال id في المكسد ب E) بالتالي أصبح المكسد $0 E$ ولمعرفة الحالة التي وصلنا لها نقوم بحساب $goto(0,E)$ (لماذا 0,E لان أول قيمتين بقيمة المكسد هي 0,E) حيث $goto(0,E)=1$ نضيف هذه القيمة للمكسد فيصبح $0 E 1$ ولأن نحن في الحالة 11 والدخل يساوي * حيث $Action(1,*)=S5$ ونتابع بنفس الخطوات

وعندما نصل الى أن قيمة المكسد $0 E 1$ والدخل فارغ أي \$ تكون الكلمة مقبولة وذلك لأن

$Action(1,\$)=Accept$

من الجدول السابق نلاحظ أن الأوتومات التي يجعل هذه الكلمة مقبولة هو :

Shift 3 -> Reduce $E \rightarrow id$ -> Shift 5 -> Shift 3 -> Reduce $E \rightarrow id$ -> Reduce $E \rightarrow E * E$ -> Shift 4 ->
 Shift 3 -> Reduce $E \rightarrow id$ -> Reduce $E \rightarrow E + E$ -> Accept



شرح المثال السابق

ملاحظة:

- عند تتبع الأوتومات السابق نلاحظ عند القيام ب Shift يمشي الأوتومات خطوة الى الامام وعند القيام ب Reduce يعود الأوتومات خطوة للخلف ثم يمشي خطوة بمسار جديد وفقاً لرقم القاعدة
- في المحاضرة القادمة سنقوم بملئ الخانات الفارغة في جدول ال LR حيث هذه الخانات مهمتها معالجة الأخطاء Error handling



" لك شيء في هذا العالم..
فقم "